

Tartu Ülikool
Loodus- ja täppisteaduste valdkond
Tehnoloogiainstituut

Robert Allik

Pmod lisamoodulite kasutamine “Digitaalse loogika” aine praktikumides

Bakalaureusetöö (12 EAP)
Arvutitehnika eriala

Juhendaja:
MSc Margus Rosin

Tartu 2019

Resümee/Abstract

Pmod lisamoodulite kasutamine “Digitaalse loogika” aine praktikumides

Antud lõputöö raames loodi kuus praktikumi juhendit “Digitaalsele loogikale”. Varasemad praktikumide juhendid on piirdunud LED-e ja lülitite väljundväärtuste muudmisega. Antud lõputöö raames lisaks eelmainitud sisend-väljund seadmetele lisati ka Pmod lisamoodulite ja nendele vastavate kommunikatsiooni protokollide kasutamise võimalus. Mainitud lisamoodulite praktikumi juhenditesse lisamisega lisandus praktikumide läbijatele parem arusaam tänapäeval kasutusel olevatest sisend-väljund seadmetest ning tõusis ka ülesannete kompleksus aste.

Kaks valminud lõputöö juhenditest võeti kasutusse ka 2019. aasta kevadsemestril Tartu Ülikoolis “Digitaalse loogika” praktikumides. Valminud praktikumi juhendi kvaliteedi hindamiseks teostati ka praktikumide läbijate seas küsitlus. Küsitluse tulemuse põhjal võib järeldada, et valminud praktikumi juhend on igati täitnud oma eesmärgi ja on praktikumide läbijatele meeldinud. Tagasisidet küsiti ka praktikumijuhendajatelt, kes samuti kiitsid valminud lõputöö selgust, ülesannete keerukust ning uudset lähenemist huvitavate lisamoodulite rakendamisel.

CERCS: T170 Elektroonika, T180 Telekommunikatsioonitehnoloogia, T190 Elektrotehnika

Märksõnad: FPGA, Basys 3, VHDL, õppematerjalid, Digitaalne loogika, Pmod

Using Pmod modules in the practicals of the “Digital logic” course

Former practical guides of the University of Tartu course “Digital logic” have been limited to using simple IO devices, such as LEDs and switches. This thesis provides six new practical guides to increase the depth of the “Digital logic” course. These new guides add the option of using Pmod modules to provide students a better understanding of modern IO devices and communication protocols.

Two of the provided guides were taught in the spring semester of 2019 in the University of Tartu course “Digital logic”. Feedback was asked of the students to measure the quality of the guides. Based on the results it can be concluded that the guides fulfilled their objectives and were liked by the students. Feedback was also asked from the practical instructors of the course and they approved of the difficulty and the new approach of these guides.

CERCS: T170 Electronics, T180 Telecommunication engineering, T190 Electrical engineering

Keywords: FPGA, Basys 3, VHDL, learning materials, digital logic, Pmod

Sisukord

Jooniste loetelu	5
Tabelite loetelu	6
Lühendid ja mõisted	7
1 Sissejuhatus	8
1.1 Probleemi tutvustus	8
1.2 Töö eesmärk ja ülevaade	9
2 Ülevaade probleemist	10
2.1 Field Programmable Gate Array	10
2.2 VHSIC Hardware Description Language ja Vivado	11
2.3 Pmod liides	12
2.4 Kommunikatsiooni protokollid	13
2.4.1 UART	13
2.4.3 SPI	13
2.4.4 PS/2	14
3 Metoodika	15
3.1 Praktikumides kasutatavad seadmed	15
3.1.1 Basys 3	15
3.1.2 Pmod GPS	17
3.1.4 Pmod AD1	18
3.1.5 Pmod AMP2	19
3.1.6 USB Human Interface Device	20
4 Tulemused ja järeldused	21
4.1 Juhendid	21
4.1.1 “Pmod GPS”	21
4.1.2 “Audio”	22
4.1.3 “USB HID”	23
4.2 Praktikumide läbijate tagasiside	25
4.3 Õpitulemuste hindamine	27
4.4 Järeldused	28
5 Kokkuvõte	29
Kasutatud kirjandus	30
Lisad	32

Lisa 1 Digitaalne Loogika Viies Praktikum - GPS	32
Lisa 2 Digitaalne Loogika Kuues ja Seitsmes Praktikum - GPS	34
Lisa 3 Digitaalne Loogika Viies Praktikum - Audio	38
Lisa 4 Digitaalne Loogika Kuues ja Seitsmes Praktikum - Audio	44
Lisa 5 Digitaalne Loogika Viies Praktikum - USB HID	48
Lisa 6 Digitaalne Loogika Kuues ja Seitsmes Praktikum - USB HID	50
Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks	54

Jooniste loetelu

Joonis 1. FPGA arhitektuur.....	11
Joonis 2. Pmod liides.....	12
Joonis 3. UART liidese kaudu saadetavad ja vastuvõetavad andmepaketid.....	13
Joonis 4. SPI liidese kaudu saadetavad ja vastuvõetavad andmepaketid.....	14
Joonis 5. PS/2 liidese ajastus diagramm.....	14
Joonis 6. Basys 3.....	16
Joonis 7. Pmod GPS.....	18
Joonis 8. Pmod AD1.....	18
Joonis 9. Pmod AMP2.....	19
Joonis 10. “Pmod GPS” juhendite loogikaskeem.....	22
Joonis 11. “Audio” juhendite loogikaskeem.....	23
Joonis 12. “USB HID” juhendite loogikaskeem.....	24
Joonis 13. “Audio” Praktikumide keerukus.....	25
Joonis 14. Üliõpilaste “Audio” teemalistes praktikumides saadud punktid.....	26

Tabelite loetelu

Tabel 1. NMEA RMC sõnumi

seletus.....17

Lühendid ja mõisted

AD1 - Pmod analoog-digitaalmuunduri moodul

AMP1 - Pmod võimendi moodul

ASIC (Application-specific integrated circuit) - Mingi kindla otstarbega kiip, mis on näiteks andmesideprotokolli, digikaamera või elektronmärgmiku jaoks projekteeritud.

CLB - Loogikaplokk, mis on üheks FPGA peamiseks koostisosaks.

FPGA - programmeeritav kombinatoorloogika-seade.

GPS - Pmod globaalse positsioneerimissüsteemi moodul

IEEE - Elektri- ja Elektroonikainseneride Instituut USA-s asuv maailma suurim erialaühing, mis toetab ülemaailmseid tehnikakonverentse, sümpoosiumeid ja seminare ning avaldab ligi 25% kõigist tehnilistest artiklitest elektrotehnika, elektroonika, arvutiehituse ja arvutiteaduse alal.

Komponent - VHDL keeles alamfaili instants

LED - Valgust kiirgav diod

LUT - Andmete tabel, kuhu on salvestatud mingi funktsiooni väärtused.

Magnetiline deklinatsioon - horisontaalne nurk geograafilise ja magnetilise põhjasuuna vahel

Pmod - Digilenti standard FPGA arendusplaatide lisamoodulitele

PS/2 - Sünkroonne jadaliides hiirte ja klaviatuuride ühendamiseks arvutiga

PWM (*Pulse width modulation*) - Pulsilaiusmodulatsioon

RAM (Random-Access Memory) - muutmälu

Slice - Loogikarakk, millest koosneb CLB

SPI (*Serial peripheral interface*) - Sünkroonne kommunikatsiooni protokoll ühe ülemseadmega

Triger - bistabiilne elektroonikalülitis

UART (*Universal Asynchronous Receiver/Transmitter*) - universaalne asünkroonne transiiver

VHDL (*Very High Speed Integrated Circuits Hardware Description Language*) - 1980.

aastatel loodud riistvarakirjelduskeel

1 Sissejuhatus

FPGA-sid kasutatakse väga paljudes erinevates valdkondades, nagu näiteks ASIC-ite prototüüpimine, video- ja pilditöötlus ning võrgundus [1]. Viimastel aastakümnetel on FPGA-de rakendamise trend on olnud pidevalt tõusev ja sellest lähtuvalt on ka oluline antud teemat Tartu Ülikoolis üliõpilastele õpetada. “Digitaalse loogika” aine raames õpetatakse üliõpilastele FPGA arhitektuuri ning FPGA-de programmeerimist VHDL keeles. Programmeeritakse Basys 3 loogikaseadet. Varasemalt on piirdutud praktikumides Basys 3 loogikaseadmetel ainult seadmel endal olevate sisend-väljund seadmete väärtuste muutmisega. Antud lõputöö peamiseks eesmärgiks on valmistada lisamoodulite praktiseerimiseks vajaliku praktikumi juhendid. Lõputöö võib lugeda positiivse tulemuse saavutanuks, kui on valminud mitmed praktikumimaterjalid ning lisaks on neist vähemalt üks ka 2019. õppeaastal üliõpilastele ette kantud ning praktikumide läbijad on mainitud praktikumimaterjali edukalt läbinud ning andnud positiivset tagasisidet.

1.1 Probleemi tutvustus

Tartu Ülikooli õppeaine “Digitaalne loogika” (ainekoodiga LOTI.05.041) eesmärk on õpetada õpilasi programmeerima FPGA-sid VHDL keeles [2] ning siiani on praktikumiülesanded piirdunud lihtsamate arvutuste ja sisend-väljund seadmete, nagu lülitid ja LED-id, kasutamisega. Selleks, et anda praktikumide läbijatele kogemusi rohkemate sisend-väljund seadmetega ja päriselule sarnaste ülesannete ning anda kasutust ülikoolil juba olemas olevatele Pmod lisamoodulitele, on vaja luua uusi õppematerjale, mis täidaks neid eesmärke. Lõputöö raames juhendas autor ka selle aine praktikume.

1.2 Töö eesmärk ja ülevaade

Bakalaureusetöö eesmärk on luua aine “Digitaalne loogika” jaoks õppematerjale, mis käsitleksid Tartu Ülikoolil juba olemasolevaid seadmeid ja nende seadmete poolt kasutatavaid kommunikatsiooni protokolle. Selleks arendas autor välja kaks praktikumijuhendit kolme erineva teema kohta.

Esimene teema põhineb “Pmod GPS” mooduli kasutamisel, millega suheldakse UART-liidese abil [3]. Teises on kasutatud SPI laadset liidest, aga ülesandeks on mõõta “Pmod AD1” mooduliga pinget, mida saadakse potentsiomeetrilt. Sellest pingest tuleb luua vastava sagedusega kastsignaali, mis edastatakse “Pmod AMP2” moodulile [4,5]. Selle mooduli kaudu on võimalik seda signaali näiteks kõrvaklappidega kuulata. Viimane teema hõlmab klaviatuurist PS/2 liidese kaudu saadud sisendi näitamist seitsme segmendilisel indikaatoril. See projekt kasutab suhtlemiseks PS/2 protokollit [6]. See küll ei kasuta ühtegi Pmod moodulit, aga annab praktikumide läbijatele kogemusi teistsuguse sisendseadmetega.

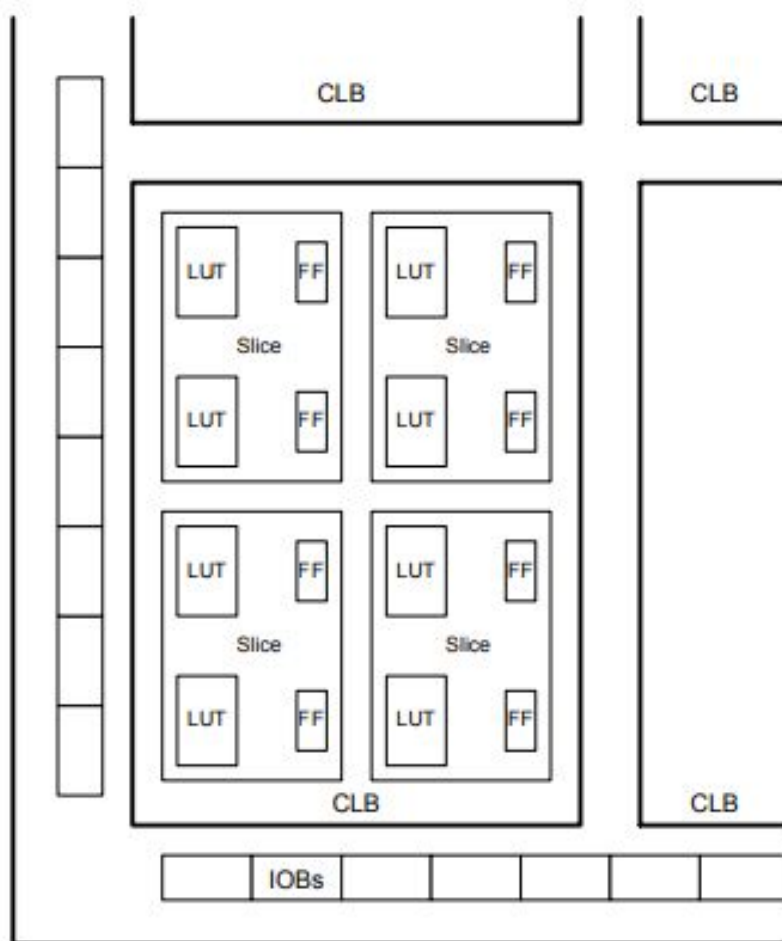
“Audio” teema juhendeid kasutati 2019. aasta kevadsemestri praktikumides. Need hõivasid kolme viimast praktikumi seitsmest ja nende kohta küsiti ka praktikumide läbijatelt tagasisidet.

2 Ülevaade probleemist

2.1 Field Programmable Gate Array

FPGA (Field Programmable Gate Array) on programmeeritav kombinatoorloogika-seade, mis koosneb konfigureeritavatest loogikaplokkidest (CLB), mille omavahelised ühendused on omakorda programmeeritavad. Iga CLB koosneb omakorda *slice*'idest, mis sisaldavad omakorda *look-up table*'eid (LUT) ja trigereid ehk *flip-flop*'id (FF). Erinevalt ASIC-itest, mis on disaini järgi toodetud, on FPGAd ka lõppkasutajal võimalik ümberprogrammeerida [1,7].

Lisaks loogikaelementidele on FPGA-des ka mälu. See sisemine mälu jaguneb kaheks: suuremahulised plokki-RAM-id ning jagatud RAM. Plokk-RAM-id on eraldi osad FPGA-st ja neid saab kasutada ainult andmete salvestamiseks ning jagatud RAM on Xilinx'i FPGA-del mäluks konfigureeritud loogikaplokkid. Erinevalt Intel'i FPGA-dest, mis kasutavad jagatud RAM-i asemel väikeseid plokki-RAM-e, on jagatud RAM ebaefektiivne elektroonikaskeemi pindala utiliseerimisel. Küll aga pakub jagatud RAM kasutajale rohkem paindlikkust [8].



Joonis 1 - FPGA arhitektuur [7]

2.2 VHSIC Hardware Description Language ja Vivado

VHDL (*Very High Speed Integrated Circuit Hardware Description Language*) on riistvarakirjelduskeel, mis määrab süsteemi käitumise. VHDL loodi USA Kaitseministeeriumi tellimusel 1980. aastatel ning see oli ka esimene riistvarakirjelduskeel, mis standardiseeriti IEEE poolt. See on mõeldud nii elektriskeemi sünteesiks kui ka selle simuleerimiseks [9].

Põhjused, miks VHDL-i kasutada, on järgmised: VHDL on standardiseeritud, see ei ole seotud ühegi FPGA tootja ega tehnoloogiaga ning on seega võimalik sama koodi kasutada erinevatel seadmetel. VHDL-i kasutatakse peamiselt programmeeritavate loogikaseadmete

programmeerimiseks ja ASIC-ite disainimiseks. See omakorda tähendab, et kui VHDLi koodis on mingi integraalskeem realiseeritud on seda lahendust võimalik implementeerida erinevatel FPGA mudelitel või toota selle järgi ASIC kiip [9].

Vivado on Xilinx'i poolt arendatud arenduskeskkond Xilinx UltraScale, seitsmenda seeria FPGA-del põhinevate süsteemide ja teiste uuemate Xilinx'i seadmete disainimiseks. Üheks FPGAde seitsmenda seeria esindajaks on ka Basys3. Vivadol on tööriistad koodi simuleerimiseks, sünteesimiseks, implementeerimiseks, seadme programmeerimiseks ja muudeks toiminguteks nagu näiteks disaini volutarbe, ressursi utiliseerimise või kiiruse efektiivistamiseks [10]. Toetatud riistvarakirjelduskeeled on VHDL, Verilog ja SystemVerilog [11].

2.3 Pmod liides

Digilenti Pmod liidest kasutatakse madala I/O nõudlikkusega välismoodulite ühendamiseks *host*-seadmete külge. Basys 3 loogikaseadmel on neli Pmod liidest. Iga selline liides koosneb kaheteistkümnest viigust. Igal liideselt on neli I/O ühendust, üks toiteühendus ja üks ühendus maanduseks [6]. Pmod'i I/O ühendusi on võimalik konfigureerida järgmisteks liidesteks: I2C, GPIO, SPI, *expanded* SPI, UART, laiendatud UART, H-sild ja kaksik H-sild [12].

Pmod liides eeldab 3,3 V toiteallikat ja seda, et signaalid vastaksid LVCMOS 3,3V või LVTTTL 3,3 V loogika standarditele. I/O ühendused on üldiselt ühendatud otse FPGA viikude külge. Pmod lisamoodulid ei eelda, et FPGA-l on *pull-up* või *pull-down* takistid, aga need võivad olla paigaldatud ka lisamoodulile endale [12].



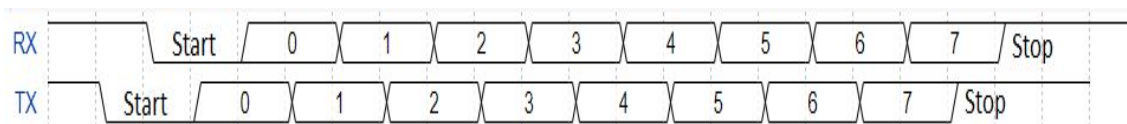
Joonis 2. Pmod liides [6]

2.4 Kommunikatsiooni protokollid

2.4.1 UART

Universaalne asünkroonne vastuvõtja/saatja (*Universal Asynchronous Receiver/Transmitter*) loodi juba 1960. aastatel Gordon Bell'i poolt Digital Equipment Corporation'is. Protokoll on universaalne selles mõttes, et saadetavate andmete formaat ja saatmise kiirus on konfigureeritavad. UART on võimeline töötama ka mitmel eri viisil: *full-duplex* ehk suhtlemine mõlemas suunas samaaegselt, *half-duplex* ehk suhtlemine mõlemas suunas aga kordamööda, ja *simplex*, mis tähendab suhtlust ainult ühes suunas. Kasutatavad viigud on TX, millega seade saadab andmed välja, ja RX, millega seade võtab andmed vastu [13].

Kommunikatsiooni alustamiseks saadab esimene seade *start*-biti, seejärel saadab ühe baidi andmeid kasutades *shift*-registrit ning lõpuks *stop*-biti. Teine seade kirjutab kätte saadud bitid SIPO (*Serial-In-Parallel-Out*) registrisse [13]. Seda liidest kasutavad “Pmod GPS” praktikumide juhendid.

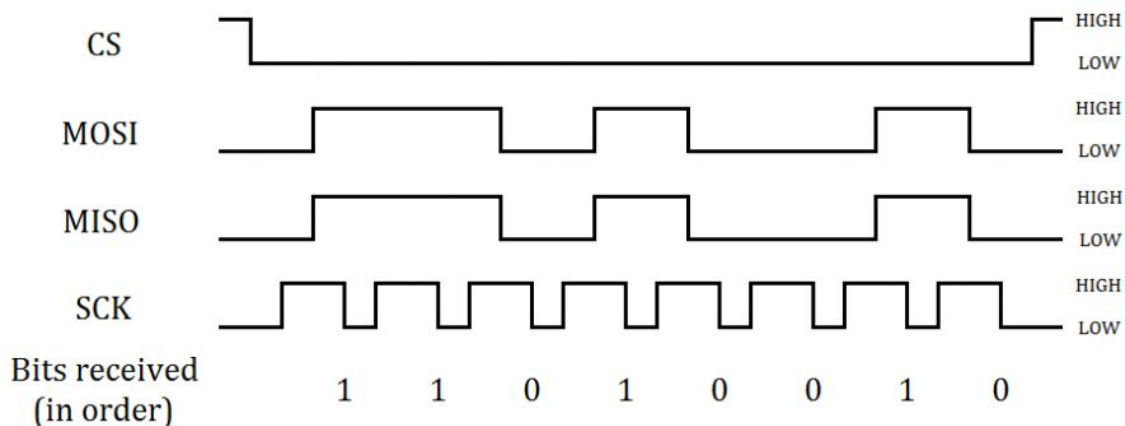


Joonis 3. UART liidese kaudu saadetavad ja vastuvõetavad andmepaketid

2.4.3 SPI

SPI (*Serial Peripheral Interface*) on telekommunikatsiooniettevõtte Motorola poolt arendatud kommunikatsiooni protokoll. SPI-l on kolm ühendust andmevahetuseks, vastav arv ühendusi alamseadme valimiseks ning toite- ja maandusühendused. Ülemseade valib CS (*Chip Select*) liinidega alamseadme ning mõlemad saavad samaaegselt infot vahetada MOSI (*Master-Out-Slave-In*) ja MISO (*Master-In-Slave-Out*) viikude kaudu. Kolmas viik on SCK (*Serial Clock*), mis määrab kiiruse, millega bitte saadetakse [14].

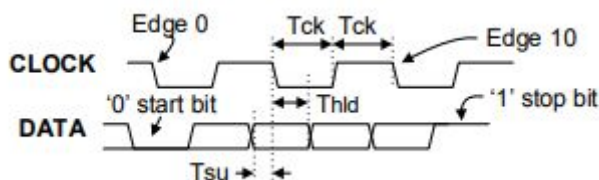
Andmevahetus algab, kui ülemseade viib vastava CS liini madalale pingele ja hoiab seda seal nii kaua, kuni andmevahetus on lõppenud. Tavaliselt saadetakse üks bitt andmeid SCK liini langeval frondil [14]. Seda liidest käsitlevad “Audio” praktikumide juhendid.



Joonis 4. SPI liidese kaudu saadetavad ja vastuvõetavad andmepaketid [14]

2.4.4 PS/2

PS/2 liides on loodud IBM'i poolt ja seda kasutatakse hiirte ja klaviatuuride ühendamiseks arvutiga. Andmete edastamiseks kasutatakse andmesignaali ja taktsignaali, mis vajavad *pull-up* takisteid, ja neid edastatakse ühe baidi haaval väikseima tähtsusega bit esimesena. Suhtlus käib üldjuhul ainult ühes suunas, alamseadmelt ülemseadmele, ja bitte loetakse taktsignaali langeval frondil [6]. Seda liidest kasutavad “USB HID” praktikumide juhendid.



Joonis 5. PS/2 liidese ajastus diagramm [6]

3 Metoodika

Juhendite loomisel seadis autor igale teemale 3 eesmärgi. Esimene eesmärk sai autoril seatud koos käesoleva töö juhendajaga tagamaks valmivate praktikumi juhendite olulisuse ja kasutatatavuse ka tulevikus. Samuti veenduti, et valitud Pmodid on endiselt toodangus tagamaks vajaduse tekkimisel nende täiendava hanke võimalikkuse. Teine eesmärk koorus välja Digitaalse loogika loengute materjalide pinnalt, milles oluline roll on massiivide, komponentide ja ka olekumasinade õpetamisel ja praktiseerimisel. Läbi loodavate praktikumide tagatakse praktikumide läbijatele veelgi parem arusaam loengus õpitust. Kolmas eesmärk tuleneb juba reaalsest vajadusest Pmodi kaasamisel disaini lahendada ära ka kommunikatsiooni protokolle küsimus.

3.1 Praktikumides kasutatavad seadmed

3.1.1 Basys 3

Praktikumides kasutatav Basys 3 loogikaseade on Digilent'i poolt toodetud arendusplaat, millel on Xilinx'i Artix 7 seeria XC7A35T-1CPG236C FPGA. Konkreetse FPGA mudeli esimene osa kirjeldab, et tegu on Xilinsi 7nda seeria Artix FPGAg, millel on ligikaudu 35 000 loogikaplokki. Teine osa nimetusest kirjeldab, et mainitud FPGA kuulub kõige aeglasemasse klassi, sellel on *wire-bond* (0.5 mm) tüüpi kiibi korpus, vastab RoHS 6/6 direktiivile erandiga 15, sellel on 236 viiku ning seade talub temperatuure vahemikus 0 °C kuni +85 °C [15].

Selles töös kasutatavad sisend/väljundseadmed on Basys 3 loogikaseadmel järgmised [16]:

- micro-B USB, mis võimaldab loogikaseadet programmeerida, koos USB-UART sillaga, mida kasutatakse osades juhendites andmete edastamiseks arvutisse;
- neli Pmod porti;
- 16 lülitit ja sama arv LED-e;
- viis vajutusnuppu;
- üks nelja numbriga seitsme segmendiline indikaator.

Basys 3 loogikaseadmel on 33 280 loogika rakku ja 5200 *slice*'i, millest igaüks omakorda sisaldab neli kuue sisendiga LUT-i ja kaheksa *flip-flop*'i [6].



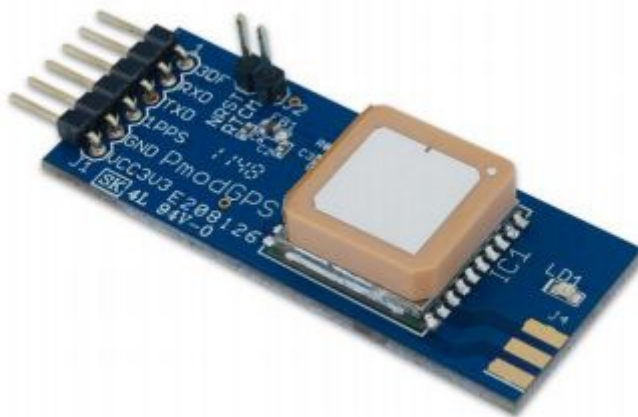
Joonis 6. Basys 3 [6]

3.1.2 Pmod GPS

Pmod GPS, mis kasutab GlobalTop FGPMOPA6H GPS moodulit, väljastab kuni 10 korda sekundis UART liidese kaudu National Marine Electronics Association (NMEA) sõnumeid. Iga sõnum algab dollari märgiga (\$), millele järgneb saatjat ja sõnumit identifitseeriv kood. Seejärel tulevad komaga eraldatud andmed. Näiteks globaalset positsiooni edastav "RMC" ehk *recommended minimum* sõnum näeb välja selline: \$GPRMC,064951.000,A,2307.1256,N,12016.4438,E,0.03,165.48,260406,3.05,W,A*55<CR><LF>.

Väli	Tähendus
\$GPRMC	Identifitseeriv kood
064951.000	UTC kellaaeg (hhmmss.sss)
A	Staatus (A = andmed on vigadeta)
2307.1256	Laiuskraad (ddmm.mmmm)
N	Põhja/lõuna indikaator
12016.4438	Pikkuskraad (ddmm.mmmm)
E	Ida/lääne indikaator
0.03	Kiirus sõlmedes
165.48	Suund kraadides
260406	Kuupäev (ddmmyy)
3.05	Magnetiline deklinatsioon kraadides
W	Ida/lääne indikaator
A	Režiim
*55	Kontrollsumma
<CR><LF>	Sõnumi lõpp

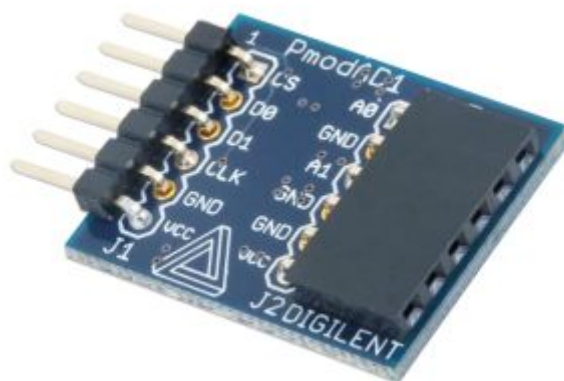
Tabel 1. NMEA RMC sõnumi seletus



Joonis 7. Pmod GPS [3]

3.1.4 Pmod AD1

Pmod AD1 on analoog-digitaalmuunduri moodul, millel on kaks 12-bitist Analog Devices AD7476A muundurit ja mõõtmisi on võimalik teha kuni miljon korda sekundis. Andmeedastus toimub SPI-laadse protokoll järgi. Erinevus tavalisest SPI-st seisneb selles, et sellel moodulil on kaks MISO signaali. See tähendab seda, et seade toimib ainult väljundina. 12 biti info edastamiseks kulub 16 taktsignaali tsüklit: esimesed neli bitti on nullid ja ülejäänud 12 tähistavad andmeid. Esimene null-bit saadetakse CS signaali langeval frondil ja teised bitid taktsignaali langeval frondil [4].



Joonis 8. Pmod AD1 [4]

3.1.5 Pmod AMP2

Pmod AMP2, millel on Analog Devices SSM2377 kiip, võimendab madala võimsusega digitaalseid või analoog sisendsignaale ja väljastab monofoonse heli. Seade võimaldab valida 6 dB ja 12 dB võimenduste vahel ja sellel on ka *pop-and-click* summutamine, mis minimiseerib müra seadme sisse- ja väljalülitamisel [17].



Joonis 9. Pmod AMP2 [5]

3.1.6 USB Human Interface Device

HID klass koosneb seadmetest, mis kasutavad USB porti ja mida kasutajad arvutisüsteemide juhtimiseks kasutavad. Näiteks HID seadmed on klaviatuurid, hiired ja muud sarnased kasutajatele mõeldud seadmed [18].

Basys 3 loogikaseadmel on Microchip'i mikrokontroller PIC24FJ128, mis annab FPGA-le USB HID ülemseadme võimekuse. Selle mikrokontrolleri ülesanne on programmeerida FPGA-d ja, kui see on tehtud, emuleerida PS/2 porti. See tähendab, et mikrokontroller käitub justkui PS/2 klaviatuur või hiir ja FPGA-l saab kasutada juba olemasolevat PS/2 liidese koodi [6].

4 Tulemused ja järeldused

4.1 Juhendid

Valminud juhendid on mõeldud asendama praeguseid Digitaalse loogika õppeaine viiendat, kuuendat ja seitsmendat praktikumi. Praktikumide läbijad õpivad nende juhendite abil kuidas kasutada massiive, olekumasinaid, erinevaid liideseid ja kuidas luua komponente. Igas juhendis on mingi osa koodist ette antud ning läbijate ülesandeks on juba olemasolevast koodist aru saada ja täita lüngad.

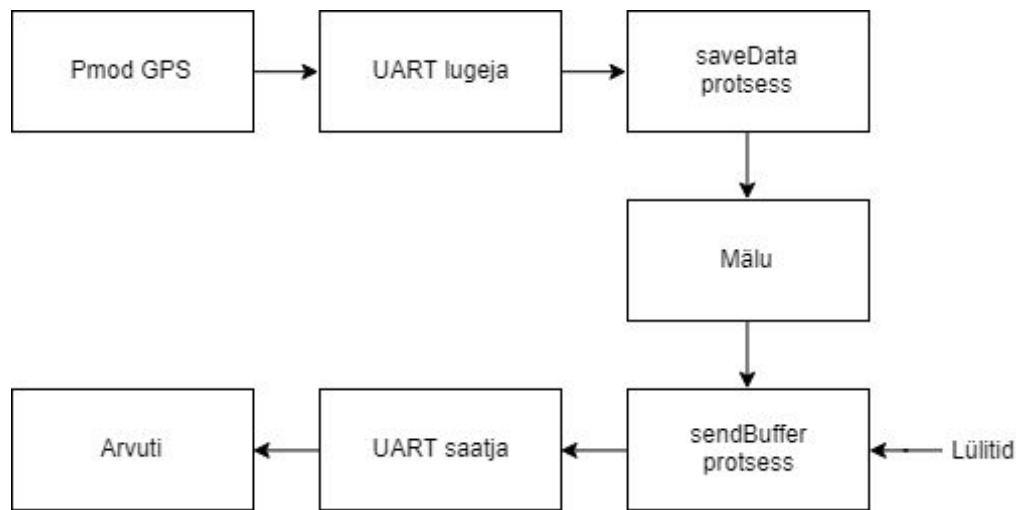
Praktikumide läbijatele antavad poolikud VHDL failid on lisatud eraldi selle tööga kaasaskäivasse tihendatud kausta.

4.1.1 “Pmod GPS”

Esimene valminud praktikumi juhend (lisa 1) annab praktikumide läbijale ülesandeks tööle saada UART liidesega andmete edastamine. Esimene praktikum on jaotatud kolmeks alamülesandeks. Esimeses alamülesandes peavad praktikumide läbijad looma olekumasinad ja defineerima nende olekud. Järgmisena peavad nad defineerima, mida andmete-saatmis-protsessis iga oleku korral programm tegema peab. Nagu igale korrektsele programmeerijale kohane järgnev programmeerimisele ka testimine. Selleks tuleb luua *constraints*-fail, kus ühendatakse sisend- ja väljundsignaalid õigete väljaviikude külge. Mainitud tegevus tagab praktikumide läbijale võimaluse oma arvutist terminali kasutades kontrollida, kas programm toimib vastavalt antud spetsifikatsioonis kirjeldatule.

Teise juhendi (lisa 2) eesmärk on täiendada olemasolevat programmi nii, et oleks võimalik salvestada GPS mooduli poolt saadetud sõnumid ja edastada nendest välja valitud osa arvutisse. Selleks peab praktikumide läbija tööle saama kaks protsessi ja looma eelmises praktikumis loodud koodist komponendi. Protsess “saveData” kirjutab UART liidese kaudu loetud baidid puhvrissi ja, kui saadi kätte vajalik sõnum, kirjutab selle mällu. Protsess

“sendBuffer” peab laskma valida lülititega, millist sõnumi osa edastatakse arvutisse ja seejärel vajalikud baidid kirjutama UART liidesesse.



Joonis 10. “Pmod GPS” juhendite loogikaskeem

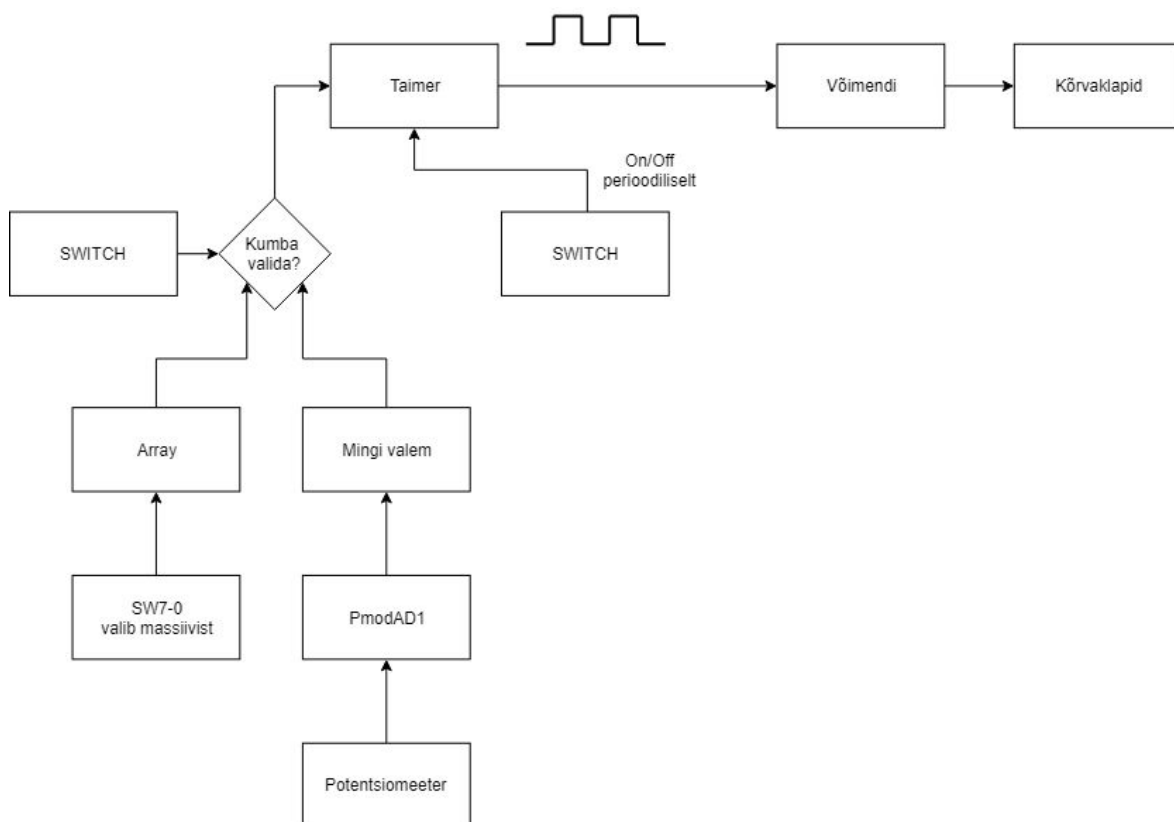
4.1.2 “Audio”

Esimeses juhendis (lisa 3) on vaja praktikumide läbijatel Pmod AD1 mooduli ja potentsiomeetriga pinget mõõta. Selles juhendis on kolm ülesannet. Esiteks peab määrama olekumasinale vajalikud olekud ja täita “timer” protsessis lüngad. Selle protsessi ülesanne on luua SPI liidese taktsignaali. Seejärel peab praktikumide läbija kirjeldama mida iga oleku korral olekumasin, mille ülesanne on vastu võtta andmeid AD1 moodulilt, tegema peab. Viimases ülesandes peab ühendama väljundsignaali LED-idega, mis laseb mõõdetud pinget lihtsasti vaadata, ja looma *constraints*-faili. Vahepealsete tulemuste kontrollimiseks on selles juhendis antud juhised simuleerimiseks ja pildid oma tulemustega võrdlemiseks.

Teise juhendis (lisa 4) peab looma kindla sagedusega signaali, mida on võimalik kõrvaklappidest kuulata. Seda võimaldab teha Pmod AMP2 moodul. Esimene ülesanne on luua massiiv, mis hoiab väärtusi, milleni loendur-protsess loeb. See protsess iga takti möödudes loendab oma väärtust üles kuni jõutakse maksimaalse väärtuseni. Maksimaalse

väärtuseni jõudmisel inverteeritakse heli tekitava signaali väärtust. Teises ülesandes peab looma võimalus lülititega valida, milline väärtus massiivist pannakse loenduri maksimaalseks väärtuseks, ning valmis saadud koodi ka mooduli endaga testida.

Lisaks nendele ülesannetele peab praktikumide läbija ka inkorporeerima eelmises praktikumis loodud koodi komponendina. Järgmisena on vaja sellest komponendist saadav arv, mis tähistab mõõdetud pinget, teisendada loendurile vastavaks arvuks praktikumide läbija enda välja mõeldud valemi abil. Lõpuks on vaja luua lisafunktsionaalsus, mis lülitab heli ühe korra sekundis sisse või välja. Selle mõtte on testida praktikumide läbija arusaama valmis saadud koodist.

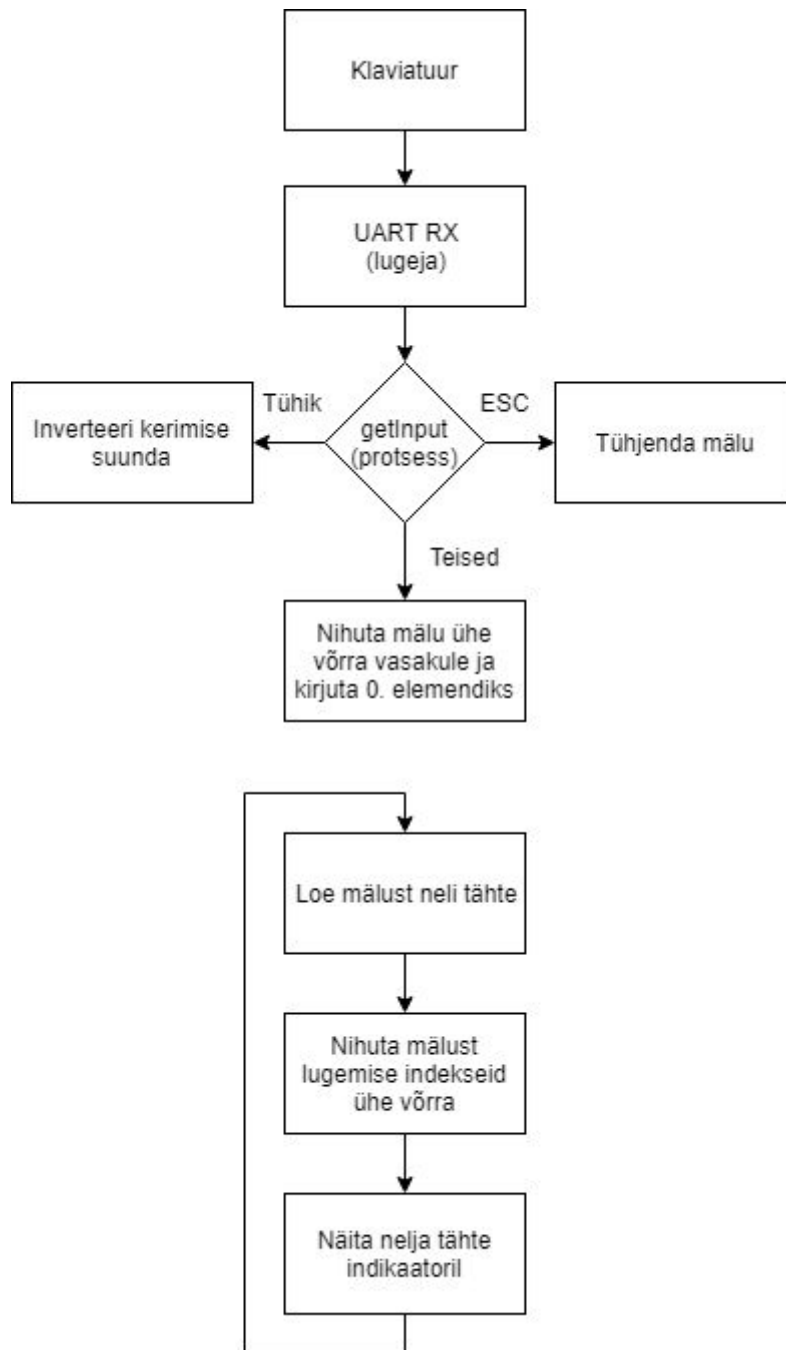


Joonis 11. “Audio” juhendite loogikaskeem

4.1.3 “USB HID”

Esimese juhendi (lisa 5) lõpuks on praktikumide läbijatel programm, mis suudab PS/2 liidese kaudu saadetud baiti LED-idel esitama. Selleks peavad nad mõistma kuidas PS/2 liides andmeid saadab ja täitma “readInput” protsessi lüngad. Tulemuse nägemiseks peavad nad sisse loetud baiti LED-idel esitama.

Teises juhendis (lisa 6) tuleb praktikumide läbijad kaasata eelnev kood komponendina ja “getInput” protsessis tuvastada klaviatuuri nupu vajutusi ja nuppu identifitseerivat kuuteistkümnendarvu tuleb näidata seitsmesegmendilisel indikaatoril. Järgmisena peavad praktikumide läbijad looma 1 Hz signaali, mille eesmärk on panna indikaatoril olevaid tähti edasi kerima. Vaja läheb ka funktsiooni, mis arvutab mälust loetavate tähtede uued indeksid. Viimase asjana peavad praktikumide läbijad lisama programmile funktsionaalsuse, mis tühiku vajutamisel muudab kerimise suunda ja “ESC” nupu vajutamisel tühjendab mälu.

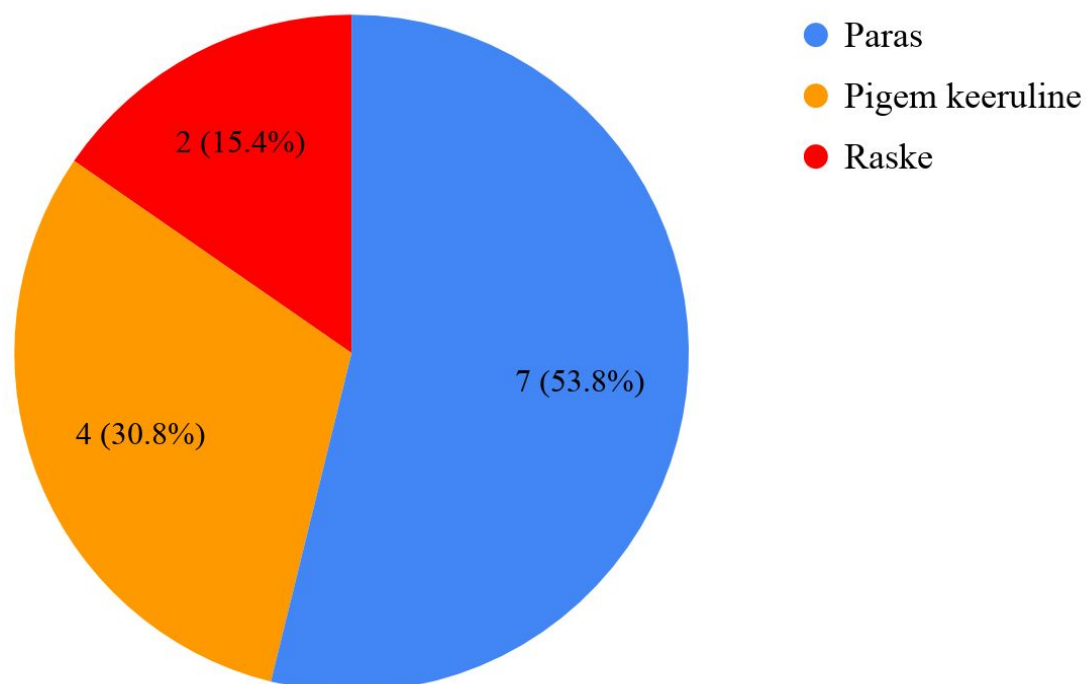


Joonis 12. “USB HID” juhendite loogikaskeem

4.2 Praktikumide läbijate tagasiside

Pärast praktikumide lõppemist küsiti praktikumide läbijatelt tagasisidet “Audio” teemaliste praktikumide kohta. Kommentaare küsiti vabas vormis ja paluti mainida juhendite keerukust, arusaadavust ning mahukust. Tagasisidet andis 13 praktikumide läbijat 27-st, kes lõpetasid

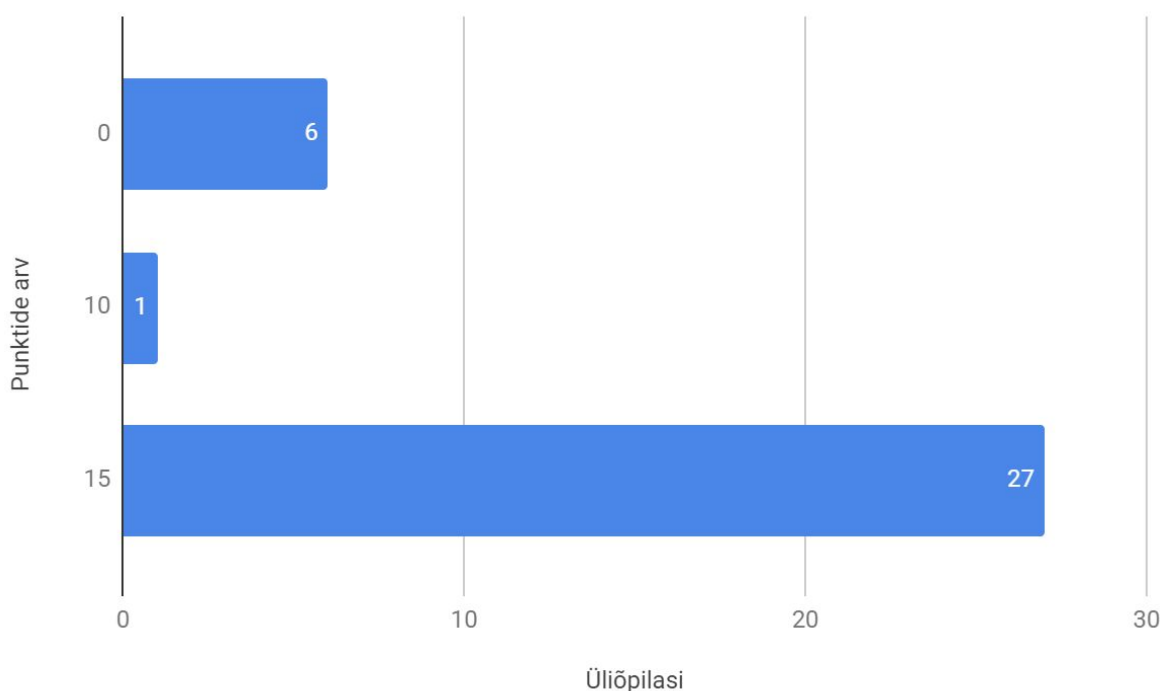
mainitud praktikumid. Peaaegu kõik vastanutest ütlesid, et praktikumid olid paraja raskusega või pigem rasked, aga juhendajate abil tehtavad (joonis 13). Lisaks sellele ei maininud ükski vastaja, et juhendid oleksid liiga mahukad ega arusaamatud. Neli vastanut mainisid, et neile meeldis juhendite progresseeruv ülesehitus ja moodulite kasutamine.



Joonis 13. "Audio" Praktikumide keerukus

4.3 Õpitulemuste hindamine

“Audio” õppematerjali läbis 34 üliõpilast. Õppematerjali raames oli kolme praktikumide peale kokku koguda 25 punkti. Need 25 punkti moodustavad “Digitaalse loogika” õppeainest 25%, seega nende mõju õpitulemustele on arvestatav. Tulemuste põhjal võime väita, et 27 praktikumide läbijat läbisid kogu õppematerjali edukalt saades kolme praktikumi tulemuste pealt kokku maksimaalse tulemuse



Joonis 14. Üliõpilaste “Audio” teemalistes praktikumides saadud punktid

Kaheksa praktikumide läbijat, kes said maksimaalsed punktid on ka teistest praktikumidest saanud maksimum tulemuse. Seega võib oletada, et tegu on väga pühendunud, õpihimuliste ja ka piisavata eelteadmistega üliõpilastega. Kõiki praktikume ei suutnud läbida seitse praktikumide läbijat. Võime sellest järeldada, et muudetud kolm viimast materjali ei sobinud neile või tekitasid piisavat raskust, vaadates nende tausta. Üldiselt võib väita, et antud juhendid ei olnud liiga keerulised ja enamus praktikumide läbijatest läbisid need maksimaalsetele punktidele.

4.4 Järeldused

Tagasiside põhjal võib väita, et “Audio” juhendid õnnestusid. Need ei ole liiga kerged, ega ka ülemäära keerulised ning täidavad kõiki autori poolt seatud eesmärgid.

Praktikumide läbijatelt kogutud tagasiside põhjal võib väita, et “Audio” juhendid täitsid oma eesmärgi. Võib järeldada, et enamikele praktikumide läbijatele valminud õppematerjal sobis. Selle läbimine nõudis kindlasti keskendumist, aga andis ka palju uusi teadmisi. Ka autor on on valminud juhendmaterjaliga rahul ning see on täitnud kõiki aegselt autori poolt seatud eesmärgid. Samuti oli juhendmaterjali kasutamine teistele teistele praktikumijuhendajatele selge ja arusaadav ning lisas ka praktikumijuhendajatele võrreldes varasema materjaliga rohkem põnevust ja valikuvabadust. Võib kindlalt väita, et lisamoodulite kasutamine peaks edaspidises “Digitaalse loogika” õppeaines kasutusse võtta ja võimalusel lisada aine raamesse ka teisi Pmod lisamooduleid. Praktikumide käigus selgus, et nagu on Basys 3 loogikaseadmete praktikumide läbijate käes vastupidavad nii on seda ka Pmod lisamoodulid.

5 Kokkuvõte

Selle lõputöö eesmärk oli luua praktikumide juhendeid ainele “Digitaalne loogika”. Uute juhendite eesmärk on anda praktikumide läbijatele kogemusi teistsuguste sisend-väljund seadmete ja erinevate kommunikatsiooni protokollidega. Samuti kasutada Tartu Ülikoolil juba olemasolevaid Pmod mooduleid.

Autor arendas välja kuus praktikumi juhendit, mis jagunevad kolmeks teemaks.

Nendeks teemadeks on “Pmod GPS”, “Audio” ja “USB HID”. “Pmod GPS” teema jaguneb omakorda kaheks juhendiks - UART protokoll ja Pmod GPS moodul. “Audio” teemalistes juhendites käsitletakse SPI protokoll ja Pmod AMP2 ning AD1 mooduleid. Need juhendid võeti ka kasutusse 2019. kevadsemestril. “USB HID” juhendite põhinevad PS/2 protokollil ja USB HID seadme kasutamisel.

Praktikumide läbijatelt saadud tagasiside põhjal võib kindlalt väita, et kasutusse võetud “Audio” praktikumijuhendid on õnnestunud ja neile seatud eesmärged täitnud.

Kasutatud kirjandus

- [1] Xilinx, What is an FPGA? Field Programmable Gate Array, [Võrgumaterjal], [tsiteeritud 11. mai 2019]. Saadaval: <https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html>
- [2] Tartu Ülikool, LOTI.05.041 Digitaalne loogika õppeaine üldandmed, [Võrgumaterjal], [tsiteeritud 2. detsember 2018], Saadaval: https://www.is.ut.ee/rwervlet?oa_aaine_info.rdf+1224119+PDF+68678017501304056762+application/pdf
- [3] Digilent, PmodGPS Reference Manual, [Võrgumaterjal], [tsiteeritud 28. oktoober 2018], Saadaval: https://reference.digilentinc.com/_media/pmod:pmod:pmodgps-gms-u1lp_rm.pdf
- [4] Digilent, PmodAD1 Reference Manual, [Võrgumaterjal], [tsiteeritud 28. oktoober 2018], Saadaval: https://reference.digilentinc.com/_media/reference/pmod/pmodad1/pmodad1_rm.pdf
- [5] Digilent, PmodAMP2 Reference Manual, [Võrgumaterjal], [tsiteeritud 10. mai 2019], Saadaval: https://reference.digilentinc.com/_media/reference/pmod/pmodamp2/pmodamp2_rm.pdf
- [6] Digilent, Basys 3 FPGA Board Reference Manual, [Võrgumaterjal], [tsiteeritud 28. oktoober 2018], Saadaval: https://reference.digilentinc.com/_media/basys3:basys3_rm.pdf
- [7] R. E. Haskell and D. M. Hanna, Introduction to digital design using digilent FPGA Boards – Block Diagram / VHDL Examples, LBE Books, Michigan, 2009.
- [8] J. P. Nicolle, FPGAs 3 - Internal RAM, [Võrgumaterjal], [tsiteeritud 11. mai 2019]. Saadaval: <https://www.fpga4fun.com/FPGAinfo3.html>
- [9] V. A. Pedroni, Circuit design with VHDL, MIT Press, Massachusetts, 2004.
- [10] Xilinx, Vivado Design Suite User Guide: Getting Started, [Võrgumaterjal], [tsiteeritud 17. mai 2019]. Saadaval: https://www.xilinx.com/support/documentation/sw_manuels/xilinx2018_1/ug910-vivado-getting-started.pdf

- [11] Xilinx, Vivado Design Suite User Guide Using the Vivado IDE, [Võrgumaterjal], [tsiteeritud 17. mai 2019]. Saadaval: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2018_1/ug893-vivado-ide.pdf
- [12] Digilent, Digilent Pmod Interface Specification, [Võrgumaterjal], [tsiteeritud 28. oktoober 2018]. Available at: https://www.digilentinc.com/Pmods/Digilent-Pmod_%20Interface_Specification.pdf
- [13] Digilent, UART, [Võrgumaterjal], [tsiteeritud 29. oktoober 2018]. Saadaval: <https://reference.digilentinc.com/learn/fundamentals/communication-protocols/uart/start>
- [14] Digilent, SPI, [Võrgumaterjal], [tsiteeritud 29. oktoober 2018], Saadaval: <https://reference.digilentinc.com/learn/fundamentals/communication-protocols/spi/start>
- [15] Xilinx, 7 Series Product Tables and Product Selection Guide, [Võrgumaterjal], [tsiteeritud 20. mai 2019]. Saadaval: <https://www.xilinx.com/support/documentation/selection-guides/7-series-product-selection-guide.pdf>
- [16] Digilent, Basys 3 Artix-7 FPGA Trainer Board: Recommended for Introductory Users, [Võrgumaterjal], [tsiteeritud 8. aprill 2019], Saadaval: <https://store.digilentinc.com/basys-3-artix-7-fpga-trainer-board-recommended-for-introductory-users/>
- [17] Analog Devices, Filterless, High Efficiency, Mono 2.5 W Class-D Audio Amplifier SSM2377, [Võrgumaterjal], [tsiteeritud 11. aprill 2019]. Saadaval: <https://www.analog.com/media/en/technical-documentation/data-sheets/ssm2377.pdf>
- [18] USB Implementers' Forum, Device Class Definition for Human Interface Devices (HID), [Võrgumaterjal], [tsiteeritud 16. mai 2019]. Saadaval: https://www.usb.org/sites/default/files/documents/hid1_11.pdf

Lisad

Lisa 1 Digitaalne Loogika Viies Praktikum - GPS

Sulle on antud fail “UART_RX_TX.vhd”, milles on olemas UART lugeja ja saatja, mis on poolik. Selles praktikumis on vaja tööle saada UART liidesega saatmine.

Puuduvad read on tähistatud kahe sidekriipsuga (--) ja rohkem infot selle kohta, kuidas see programm töötab, leiad juhendi lõpust.

Ülesanne 1: Esiteks on vaja luua olekumasinad “current_state_” ja määrata vajalikud olekud.
(2 punkti)

Ülesanne 2: Protsessis “transmit” määra, mida iga oleku korral tehakse.
(4 punkti)

Ülesanne 3: Et seda saaks testida on vaja kontrollida signaale “i_TX_enable” ja “internal_TX_byte”. Esimest saab otse lülitiga juhtida, aga teise jaoks on vaja luua loogika, mis muudab vastavalt lülititele 7 - 0 “internal_TX_byte” signaali. Näiteks kui lüliti 0 on sees, siis “internal_TX_byte” = “00001111”. Testimise ajaks tuleb ka “transmit” protsessis siis seda signaali kasutada.

Nüüd loo *constraints* fail, kus

- i_clk : taktsignaali
- i_TX_enable : lüliti 15
- o_TX_serial : J4 USB RX
- i_TX_byte : lülitid 7 - 0
- RX signaalid ühenda samuti LED-ide ja lülititega

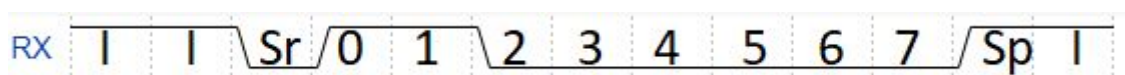
Ava oma arvutis terminali programm (Termite, Realterm) ja vaata, kas sa saad õiged tähed kätte.

(4 punkti)

Kokku 10 punkti.

UART

Universaalne asünkroonne vastuvõtja/saatja (*Universal Asynchronous Receiver/Transmitter*) loodi juba 1960. aastatel. Kasutatavad liinid on TXD, millega seade saadab andmed välja, ja RXD, millega seade võtab andmed vastu. *Baud rate* määrab kui palju bitte (kaasa arvatud *start*-ja *stop*-bitid) saadetakse ühes sekundis, Selles praktikumis kasutatav UART on 8 andme-bitiga, ilma paarsuskontrollita ja ühe *stop*-bitiga.



Joonis 1. UART liidese kaudu saadakse bait “00000011” (LSB esimesena). I tähistab *idle*-olekut, Sr on *Start*-bit, siis andme-bitid ning Sp on *Stop*-bit.

Protsess “transmit” toimib nii, et iga biti pikkus on kindel arv taktsignaali tsükleid. Näiteks kui *baud rate* on 9600, siis ühe biti pikkus on 10417 taktsignaali tsükli. Andmeedastust alustab “TX_enable” signaali tõusev front. See tähendab seda, et iga baidi saatmiseks peab seda signaali viima madalast olekust kõrgeks. Esiteks saadetakse *start*-bit, siis andme bitid ja lõpuks üks *stop*-bit. Näiteks *start*-biti saatmisel kirjutatakse “o_TX_serial” madalaks ja hoitakse seda sellisena kuni lugeja on jõudnud 10416-ni. Järgmisel tsüklil hoitakse seda signaali ikka madalana aga olekut muudetakse ning kokku tuleb ikkagi 10417 tsükli.

“Receive” protsess toimib sarnaselt, aga bitte loetakse sisse 10417/2 tsükli ajal, ehk siis biti keskelt.

Lisa 2 Digitaalne Loogika Kuues ja Seitsmes Praktikum - GPS

Selles praktikumis on vaja salvestada GPS mooduli poolt saadetud sõnumid ja edastada nendest valitud osa arvutisse.

Ülesanne 1: Protsessi “saveData” mõte on salvestada Pmod GPS-ilt saadud baite puhvrissse ja kirjutada puhvriss see vastavasse signaali. Näiteks kui GPS saadab RMC sõnumi, siis kirjutatakse puhvrissse `bff(15, 15 downto 10)=$GPRMC`. Siis me kontrollime, kas `bff(15, 15 downto 10)` olevad baidid võrduvad `$GPRMC`. Kui jah, siis kirjutame selle “rmc” signaali. Kontrollimiseks ühenda näiteks `bff(15,15)` LED-idega, et näha selle ASCII koodi. See peab alati olema dollari märk \$.

(6 punkti)

Ülesanne 2: Protsess “sendBuffer” saadab välja valitud rmc signaali väljad UART liidese kaudu arvutisse. See on kahe olekuga olekumasin. Esimeses olekus hoitakse UART saatjat väljalülitatuna ja laseb lülitega valida, millist osa sõnumist arvutisse saata. Teises olekus kirjutatakse “message” muutujas olevad baidid UART saatjasse.

(6 punkti)

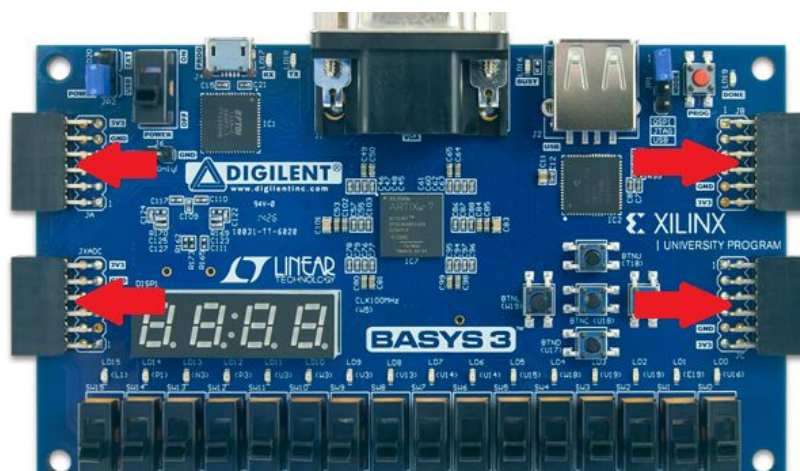
Ülesanne 3: Loo võimalus ka ühte teist sõnumit vastu võtta. Selleks peab looma uue signaali selle hoidmiseks, muutma “saveData” protsessis puhvriss kirjutamist ja “sendBuffer” protsessis “message” muutuja kirjutamist.

(3 punkti)

Kokku 15 punkti.

Pmod liides

Digilenti Pmod liides on mõeldud väikese I/O nõudlikkusega välismoodulite ühendamiseks *host*-seadme külge. Basys 3-el on neli Pmod porti (need 2x6 mustad konnektorid). Igal real on 3.3V, GND ja 4 I/O ühendust. Need liidesed lasevad meil kergesti ühendada erinevaid lisamooduleid.



VCC	GND	4	3	2	1
VCC	GND	10	9	8	7

Joonis 1. Pmod port

JA	JB	JC	JXADC
JA1: J1	JB1: A14	JC1: K17	JXADC1: J3
JA2: L2	JB2: A16	JC2: M18	JXADC2: L3
JA3: J2	JB3: B15	JC3: N17	JXADC3: M2
JA4: G2	JB4: B16	JC4: P18	JXADC4: N2
JA7: H1	JB7: A15	JC7: L17	JXADC7: K3
JA8: K2	JB8: A17	JC8: M19	JXADC8: M3
JA9: H2	JB9: C15	JC9: P17	JXADC9: M1
JA10: G3	JB10: C16	JC10: R18	JXADC10: N1

Pmod GPS

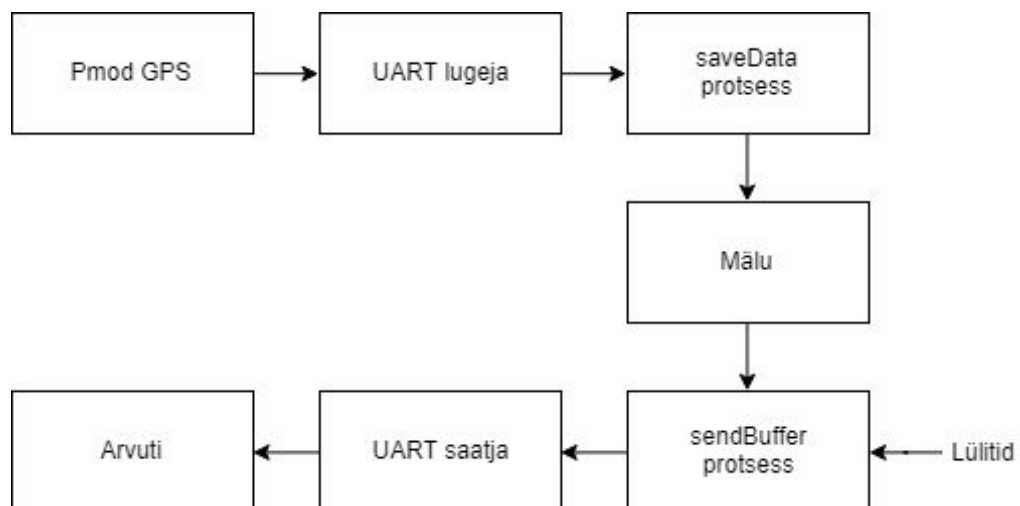
Pmod GPS moodul saadab kuni 10 korda sekundis UART liidese kaudu National Marine Electronics Association (NMEA) sõnumeid, mida on kokku sellel moodulil viis. Iga sõnum algab dollari märgiga (\$), millele järgneb saatjat identifitseeriv kood (GP) ja sõnumi nimi. Seejärel tulevad komaga eraldatud andmed. Moodulil on ka vilkuv LED, mis näitab, kas ühendus on satelliitidega saavutatud. Kui LED vilgub, siis ühendust ei ole. Ühenduse saavutamine võib võtta paar minutit aega.

Selles praktikumis kasutame RMC sõnumit, mis näeb välja selline:

\$GPRMC,064951.000,A,2307.1256,N,12016.4438,E,0.03,165.48,260406,3.05,W,A*55

Väli	Täheendus
\$GPRMC	Identifitseeriv kood
064951.000	UTC kellaaeg (hhmmss.sss)
A	Staatus (A = andmed on vigadeta)
2307.1256	Laiuskraad (ddmm.mmmm)
N	Põhja/lõuna indikaator
12016.4438	Pikkuskraad (ddmm.mmmm)
E	Ida/lääne indikaator
0.03	Kiirus sõlmedes
165.48	Suund kraadides
260406	Kuupäev (ddmmyy)
3.05	Magnetiline deklinatsioon kraadides
W	Ida/lääne indikaator
A	Režiim
*55	Kontrollsumma
<CR><LF>	Sõnumi lõpp

Rohkem infot mooduli ja selle sõnumite kohta leiab selle [reference manual](#)-ist.



Joonis 2. Loogikaskeem

Lisa 3 Digitaalne Loogika Viies Praktikum - Audio

Selles juhendis on sulle antud fail “ad1.vhd”. See fail on vigane ning osad read on kaduma läinud või poolikud. Keegi ei mäleta enam, kuidas see töötab ning sinu ülesanne on see uuesti tööle saada ja mõõta Pmod AD1 mooduli ja potentsiomeetriga pinget.

Puuduvad read on tähistatud kahe sidekriipsuga (--) ja rohkem infot selle kohta, kuidas see programm töötab, leiab juhendi lõpust.

Sulle on antud ka .bit fail, mis näitab kuidas programm peab lõpuks töötama. See kasutab 13. lüliti ja Pmod JC ülemist rida

Ülesanne 1: Esimesena tuleb olekumasinale määrata olekud ja “timer” protsess on vaja tööle saada. Kui olekumasin ei ole oote olekus, siis peab “counter”-it suurendama ühe võrra ja kui on jõutud lõppväärtuseni tuleb väljund taktsignaali inverteerida ning loendamist uuesti alustada.

(3 punkti)

Ülesanne 2: Protsessis “main” määra, mida iga oleku korral tehakse. Kontrolli seda ja eelmist ülesannet simuleerimisega.

(3 punkti)

Ülesanne 3: Lõpuks tahame mõõdetud pinget LED-idel näidata. Selleks tuleb “buffer0” ühendada “o_output0” väljundsignaali. Loo *constraints* fail ja ühenda sisend- ja väljundsignaalid õigete viikude külge.

- “o_output0” ühenda LED-idega
- “i_enable” on lüliti
- “clk” on taktsignaali
- Ülejäänud on Pmod port

Loo .bit fail, lae see Basys 3-ele.

(4 punkti)

Kokku 10 punkti.

Pmod liides

Digilenti Pmod liides on mõeldud väikese I/O nõudlikkusega välismoodulite ühendamiseks *host*-seadme külge. Basys 3-el on neli Pmod porti (need 2x6 mustad konnektorid). Igal real on 3.3V, GND ja 4 I/O ühendust. Need liidesed lasevad meil kergesti ühendada erinevaid lisamooduleid.



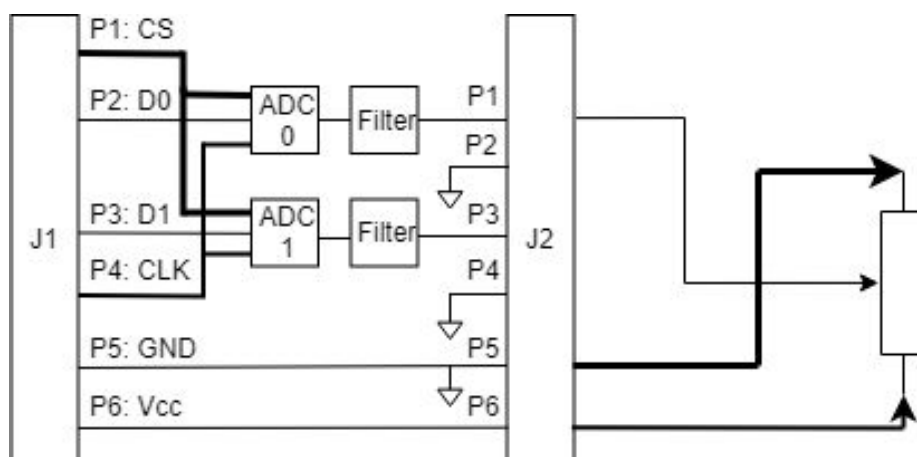
VCC	GND	4	3	2	1
VCC	GND	10	9	8	7

Joonis 1. Pmod port

JA	JB	JC	JXADC
JA1: J1	JB1: A14	JC1: K17	JXADC1: J3
JA2: L2	JB2: A16	JC2: M18	JXADC2: L3
JA3: J2	JB3: B15	JC3: N17	JXADC3: M2
JA4: G2	JB4: B16	JC4: P18	JXADC4: N2
JA7: H1	JB7: A15	JC7: L17	JXADC7: K3
JA8: K2	JB8: A17	JC8: M19	JXADC8: M3
JA9: H2	JB9: C15	JC9: P17	JXADC9: M1
JA10: G3	JB10: C16	JC10: R18	JXADC10: N1

PmodAD1

PmodAD1 on ADC (*analog-to-digital converter*), mille suhtlus Basys 3-ga käib SPI-laadse protokolliga. Erinevus tavalisest SPI-st tuleneb sellest, et andmete edastamine käib ainult ühte pidi. ADC tagastab 12 bitse märgita arvu ($1\text{LSB} = V_{\text{dd}}/4096$, kus $V_{\text{dd}} = 3.3\text{V}$ ehk toitepinge). **Juhtmed ühenda järgmiselt: J2P1 ühenda potentsiomeetri keskmisele jalale, J2P5 ja P6 ühenda teiste jalgade külge.**



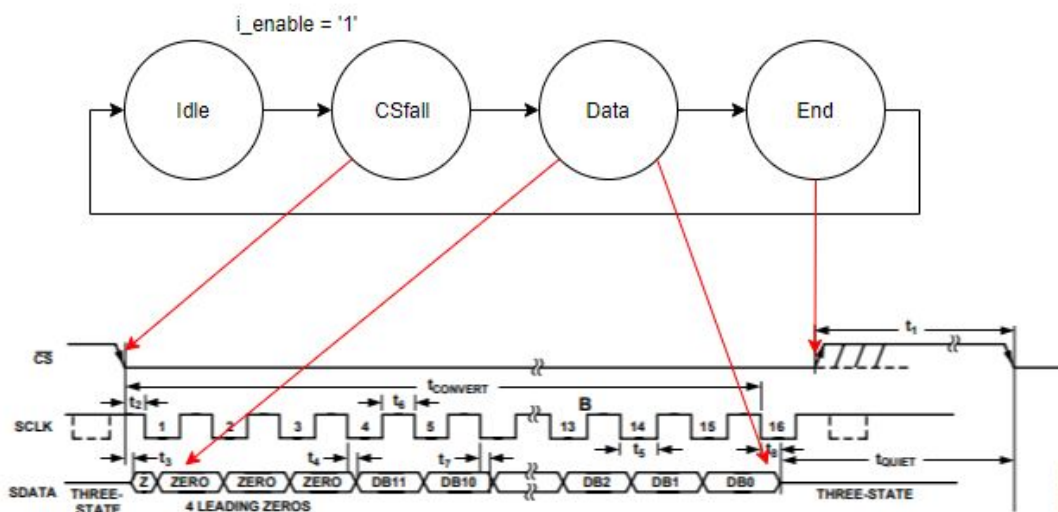
Joonis 2. Pmod AD1 skeem ja potentsiomeeter

PmodAD1 protsesside seletused

Koodist on lihtsam aru saada, kui samal ajal vaadata AD7476A *timing diagram*-i (skeem 2).

timer : process tekitab 10 kHz taktsignaali siis, kui mõõtmine lülitatakse sisse. Esimene SCLK takt algab kõrge oleku keskelt.

main : process on olekumasin. **s_idle** olekus oodatakse kuni lülitiga lülitatakse mõõtmine sisse, siis **s_CSfall** olekus CS signaal läheb '1' -> '0'. **s_data** olekus kirjutatakse üks bitt "data0" vektorisse ning vähendatakse indeksit, sest suurima tähtsusega bit saadakse esimesena. Kui kõik bitid on käes, siis minnakse **s_end** olekusse, vektor kirjutatakse puhvrisse ja CS signaal läheb uuesti kõrgeks.

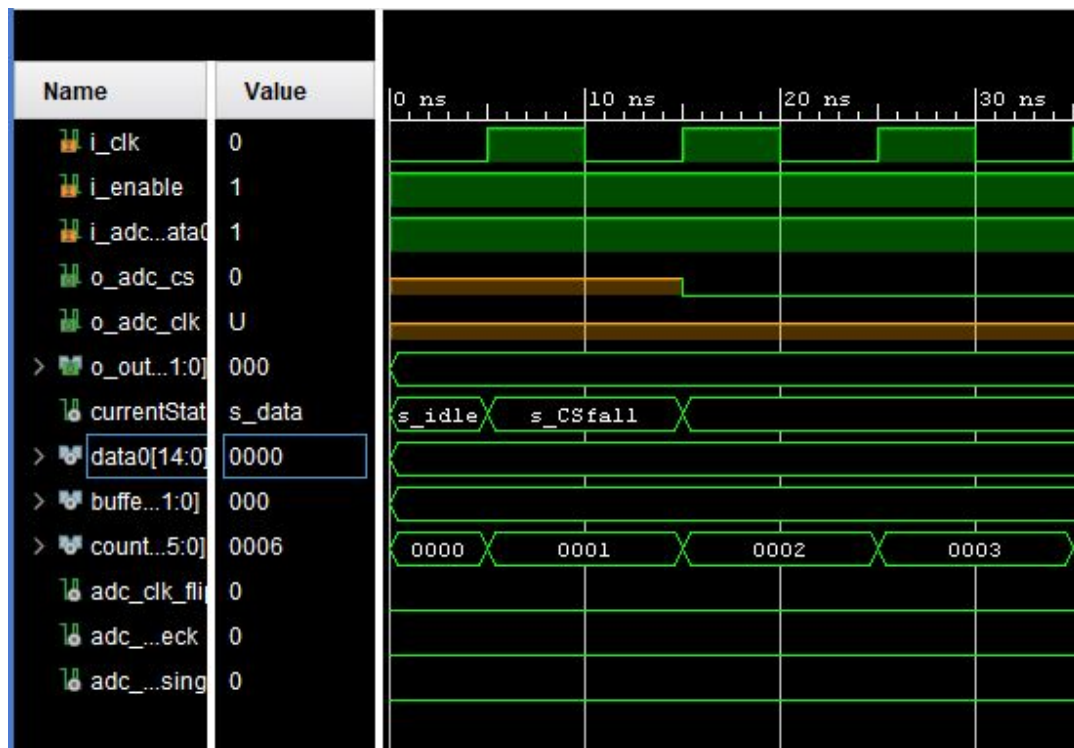


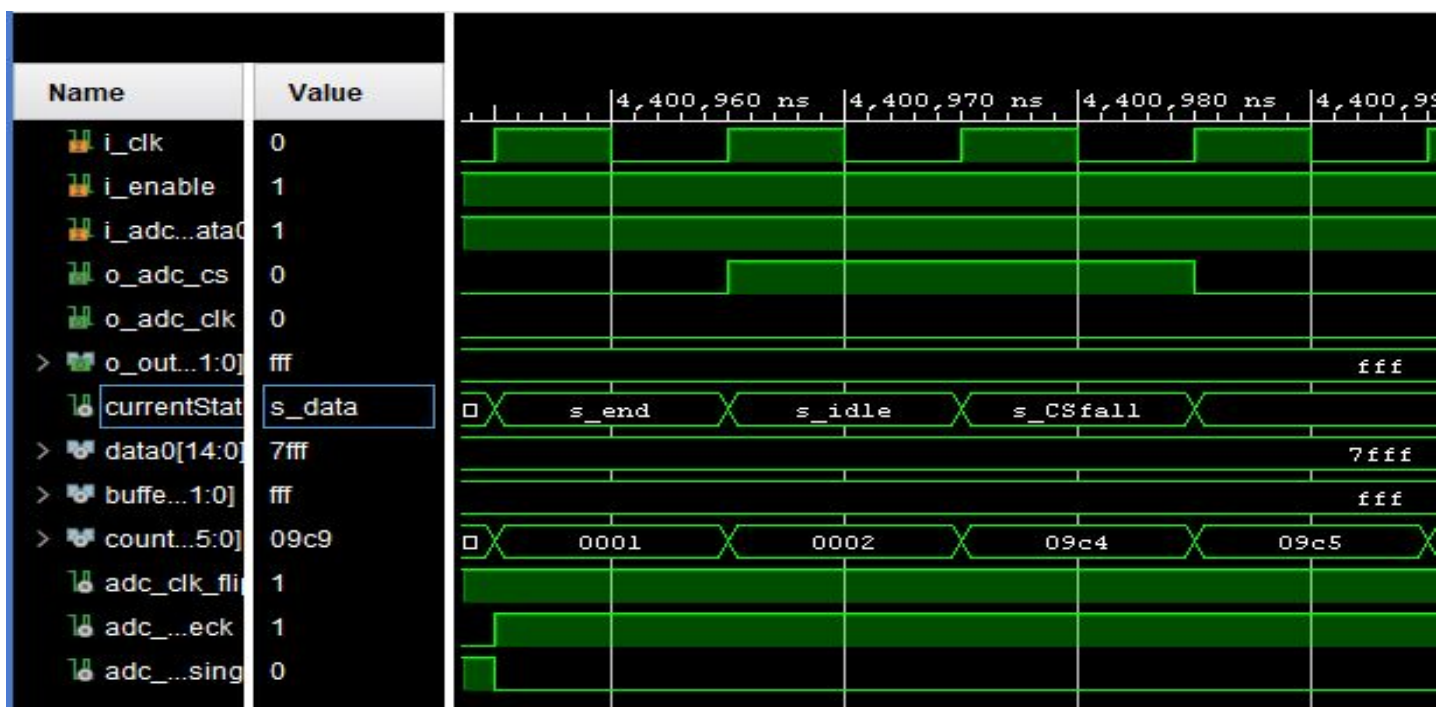
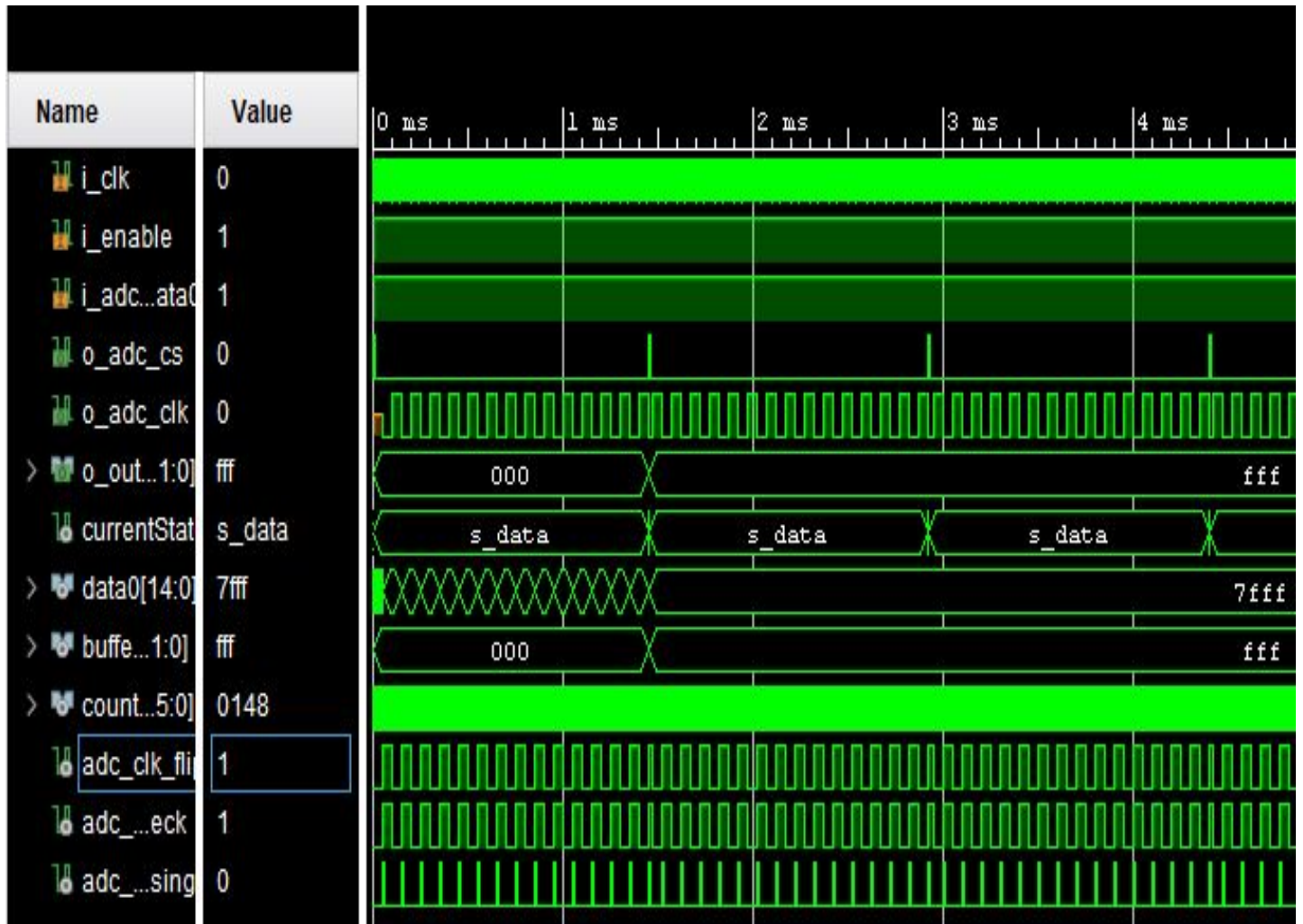
Joonis 3. Pmod AD1 olekumasin ja AD7476A *timing diagram*

Simuleerimine

Vali **Run Simulation** -> **Run Behavioral Simulation**, siis määra järgmised sisendid:

- **i_clk** -> **Force Clock** 10 ns perioodiga
- **i_enable** -> **Force Constant 1**
- **i_adc_data0** -> **Force Constant 1**





Lisa 4 Digitaalne Loogika Kuues ja Seitsmes Praktikum - Audio

Teises osas on sulle antud fail “main.vhd”. See fail on samuti poolik. Selle praktikumi lõpuks peab Pmod AMP2 tekitama kõrvaklappidega heli ja peab olema võimalik heli sagedust lülititega valida.

Ülesanne 1: Täida “TOP_values” massiiv arvudega, mis vastavad kindlatele sagedustele.

Näiteks kui sisestada arv 50 000 000, siis heli sagedus oleks 1 Hz, sest seadme enda taktsignaali on 100 MHz.

Nüüd on vaja tööle saada “timer” protsess. Praegu peab see iga taktiga “counter” signaali suurendama ja kui on jõutud TOP väärtuseni, siis tuleb “audio_out” signaali inverteerida.

Arvesta ka olukorda, kui TOP seatakse väiksemaks kui “counter” signaal.

Ülesanne 2: Protsessis “main” täida praegu ainult “else” osa. Vaata, milline lüliti on sisse lülitatud, ja kirjuta vastav väärtus “TOP_values” massiivist “counterTOP” signaali. See laseb meil lülititega valida, millise sagedusega heli tekitatakse.

Ülesanne 3: Viimase asjana on vaja lülititega juhtida “gain” ja “shutdown” väljundeid. AMP2 ühenda Pmod JB alumisse ritta. **Tekitav heli on arvatavasti väga vali, ole valmis lülitiga seda välja lülitama.** Loo .bit fail, lae see Basys 3-ele.

(8 punkti)

Seitsmes Praktikum

“main.bit” fail näitab kuidas programm peab lõpuks töötama.

- AMP2 => Pmod JB alumine rida
- AD1 => Pmod JC ülemine rida
- switch(7 downto 0) => valib massiivist TOP väärtuse
- switch(8) => valib ADC ja lülitite sisendite vahel
- switch(9) => AMP2 gain
- switch(10) => AMP2 shutdown
- switch(13) => AD1 enable
- switch(15) => lülitab heli perioodiliselt sisse/välja

Ülesanne 4: Nüüd tahame ühendada eelneva ADC koodi praeguse koodiga. Selleks loo “main.vhd” failis ad1 komponent ja selle *port map*. Ühenda ad1 “o_output0” signaal “led” signaaliga.

(2 punkti)

Ülesanne 5: “main” protsessis, kui 8. lüliti on sees, siis loe muutujasse “x” ADC väljund. Mõtle välja lineaarne funktsioon ($y = a * \text{ADC} + b$), mis teeb ADC väljundi väärtustest vastavad väärtused meie “timer”-ile. Näiteks kui ADC tagastab arvu 0, siis tekitatakse kõige madalama sagedusega heli, ja kui tagastatakse arv “1111111111”, siis kõige suurema sagedusega heli.

(3 punkti)

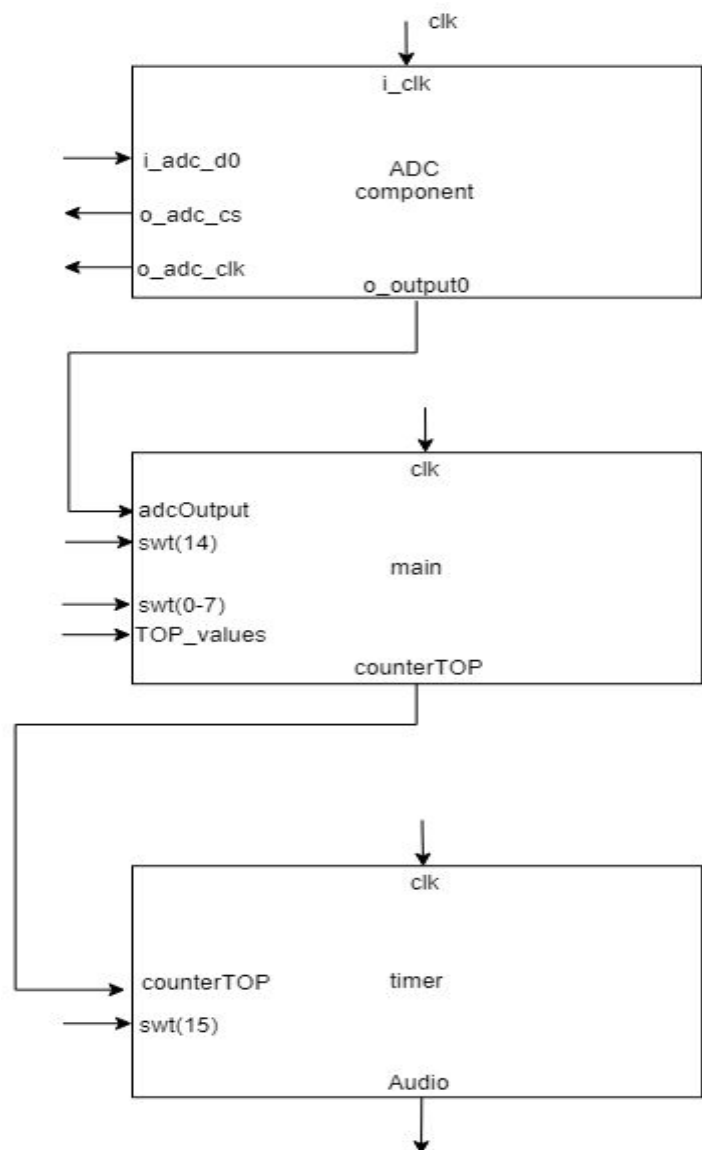
Ülesanne 6: Loo lisa funktsionaalsus, mis lülitab korra sekundis heli sisse/välja. “Timer” protsess peab suurendama “counter_oneSecond” signaali ja kui see on jõudnud õige arvuni ja vastav lüliti on sees, siis invertteeritakse “soundOn” signaal. See signaal peabki lülitama heli välja.

(2 punkti)

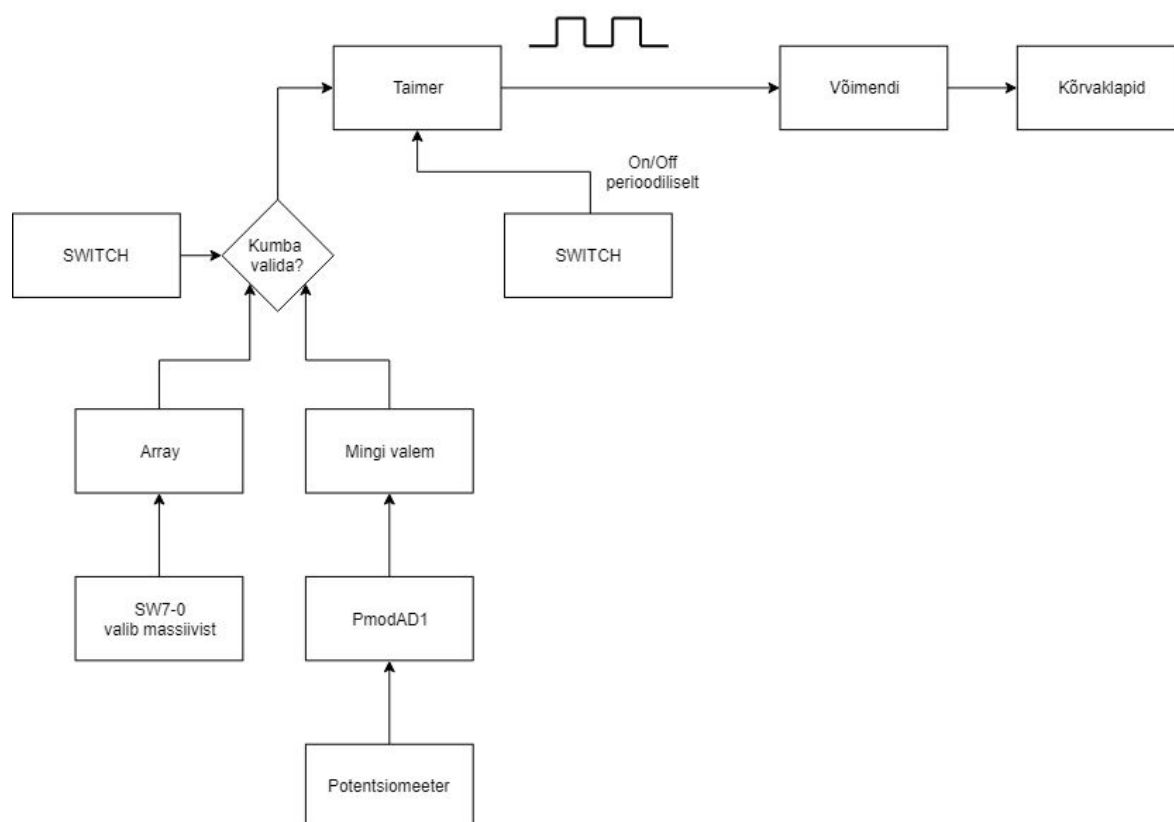
Kokku 15 punkti.

Pmod AMP2

See lisamoodul võimendab Basys 3-elt tulevat signaali nii, et seda on võimalik kõrvaklappidest kuulata. Sisendiks võib olla nii PWM kui ka analoog signaal. 3.5mm konnektor on mõeldud mono heli edastamiseks. Selleks, et heli tekitada on vaja genereerida AMP2-he “Audio Input” *pin*-ile signaal. Seadmel on ka võimalik valida 12 dB või 6 dB võimendus ja on olemas väljalülitussignaali.



Joonis 1. Protsesside vahelised signaalid



Joonis 2. Loogikaskeem

Lisa 5 Digitaalne Loogika Viies Praktikum - USB HID

Selle ja kahe järgmise praktikumi eesmärk on luua selline programm, mis loeb klaviatuurilt vajutatud nuppe ja kirjutab need mälli. Tekst kuvatakse seitsmesegmendilisel indikaatoril ja seda on võimalik edasi ja tagasi kerida. Selles praktikumis on vaja tööle saada klaviatuurilt nuppude vajutamise lugemine.

Ülesanne 1: Faili “USB_HID_reader.vhd” ülesandeks on lugeda sisse andmeid, mida klaviatuur saadab. Selleks määra vajalikud sisend/väljund signaalid (i_clk, ps2_clk, ps2_data, o_done ja o_output). Sisemise signaalina on vaja ainult ühte õige pikkusega vektorit, mis toimib puhvrina.

Meil on vaja signaalidest i_clk ja i_ps2_clk sõltuvat protsessi “readInput”, see protsess toimib olekumasinana ja sul on vaja defineerida olekumasina signaal. Protsess loeb iga ps2_clk langeva frondiga sisse ühe biti andmeid ja kirjutaks selle puhvrisse. Kui kõik bitid on sisse loetud, siis on vaja puhvrist üks bait väljundisse kirjutada. Väljund o_done määratakse i_clk tõusva frondiga ja see on ‘1’ siis, kui kirjutatakse puhvrist väljundisse, muidu on ‘0’. Sõltuvus i_clk signaalist on tähtis sellepärast, et see hoiab seda programmi osa sünkroonis ülejäänud programmiga. On antud ka simuleermise fail. Simuleerimise tulemus peab olema sama nagu 1. joonisel.

(8 punkti)

Ülesanne 2: Näita sisse loetud baiti LED-idel. PS/2 liidesel on vaja *pull-up* takisteid, selleks peab *constraints* failis kasutama järgmiseid ridu:

```
set_property -dict {PACKAGE_PIN C17 IOSTANDARD LVCMOS33 PULLUP true}
[get_ports i_ps2_clk];
set_property -dict {PACKAGE_PIN B17 IOSTANDARD LVCMOS33 PULLUP true}
[get_ports i_ps2_data];
```

(2 punkti)

Kokku 10 punkti.

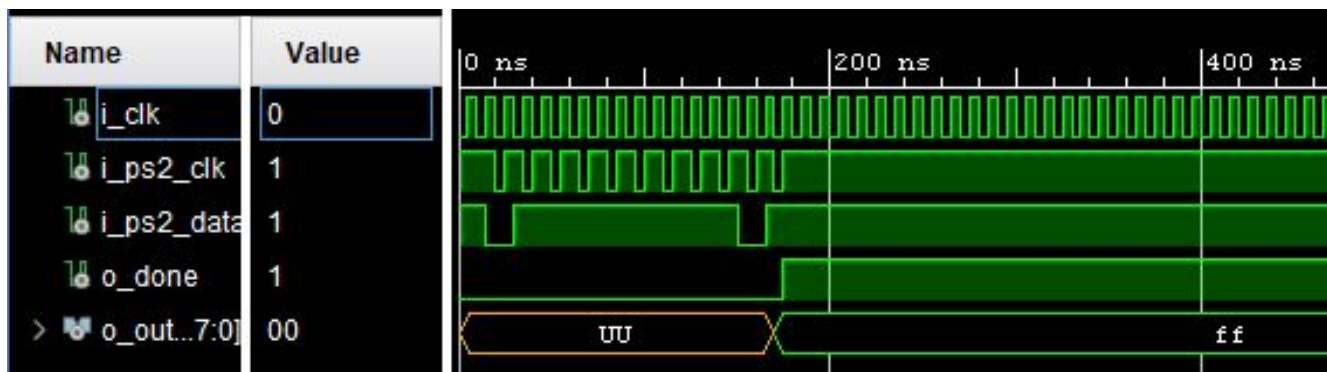
USB - Human Interface Device

Human Interface Device-id on seadmed, mis võimaldavad inimesel arvutit juhtida, nagu näiteks klaviatuurid ja hiired. Basys 3-e USB-A port on ühendatud mikrokontrolleri külge, mis emuleerib PS/2 liidest. Emuleerimist tehakse selle pärast, et uued programmid saaksid ära kasutada juba olemasolevat loogikat (*IP Core*), aga kuna see liides on suhteliselt lihtne saame me selle loogika ka ise kirjutada.

PS/2 liides

Liides kasutab kahe ühendusega jadasiini (taktsignaali ja andmesignaali). Andmesignaali kaudu saadetakse 11 biti, milleks on start bit, andmebitid (*least significant bit* esimesena), *parity* ja stop bit, ning igat bitti peab sisse lugema liidese taktsignaali langeval frondil.

Klaviatuuri korral saadetakse nupu vajutusel kolm baiti. Näiteks kui vajutatakse nuppu A, siis saadetakse bait "1C" ja kui see nupp lastakse lahti saadetakse järjest "F0" ja "1C". Nuppude tabeli leiab [Basys 3 Reference Manual](#)-ist.



Joonis 1. PS/2 liidese simuleerimine

Lisa 6 Digitaalne Loogika Kuues ja Seitsmes Praktikum - USB HID

Kui klaviatuurilt on võimalik nupuvajutusi tuvastada, siis on vaja luua loogika, mis nupuvajutused mällu kirjutab ja neid seitsmesegmendilisel indikaatoril näitab.

Ülesanne 1: Failis “main.vhd” loo USB_HID_reader-ile *komponent* ja *port map*. Need lasevad kaks eraldi vhdl faili üheks programmiks ühendata.

Protsess “getInput” loeb sisse ühe baidi siis, kui on taktsignaali tõusev front ja “read_complete_re” = ‘1’. See bait konverteeritakse täisarvuks ja kirjutatakse muutujasse “input_holder”. Seejärel vaadatakse, kas eelmine saadetud bait oli nupu lahti laskmine. Kui jah, siis kirjutatakse muutuja “key_released” nulliks ning mälu nihutatakse vasakule ning see bait kirjutatakse esimeseks elemendiks.

(3 punkti)

Ülesanne 2: Näita seitsmesegmendilisel indikaatoril viimase lahti lastud nupu kuueteistkümnendsüsteemi koodi. Selleks on vaja protsess “display2” 14. lüliti olek täita. Kuueteistkümnendarv on vaja kümnendsüsteemi ümber arvutada.

(3 punkti)

Seitsmes praktikum

Nüüd tahame luua järgmised funktsionaalsused: mälust näidatavate tähtede kerimine, kerimise suuna muutmine tühiku vajutamisel ja “ESC” nupu vajutamisel mälu tühjaks tegemine. Jooniselt 1 näeb protsesside vahelisi signaale.

Ülesanne 1: Mälu kerimiseks on esiteks vaja “timer” protsessis teha veel üks taktsignaali jagaja “clkdiv1” ja muuta sellega “cclk1” signaali. See signaal peab muutuma sagedusega 1 Hz.

Protsessis “display1” kutsub välja funktsiooni “move_display” uute indeksite arvutamiseks ja kirjutab vastavalt nendele massiivi “disp_array” uued arvud. Seda tehakse signaali “cclk1” tõusva frondi korral.

(3 punkti)

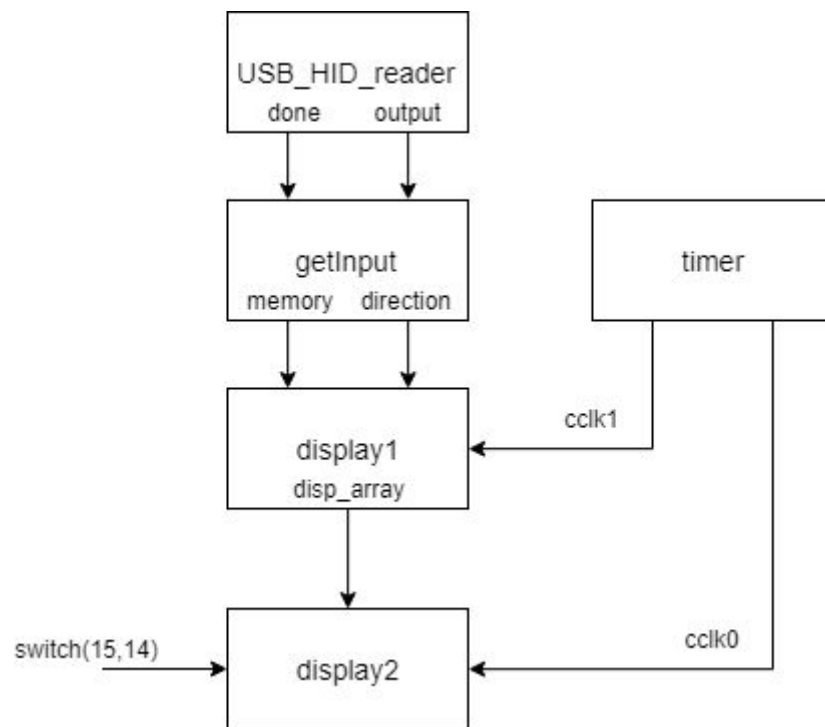
Ülesanne 2: Loo funktsioon “move_display”. See võtab sisenditeks ühe täisarvude massiivi, mis hoiab indekseid, ja “direction” signaali. Vastavalt “direction” signaalile massiivi sees olevaid arve suurendatakse või vähendatakse ühe võrra. Tähelepanu tuleb ka pöörata olukorrale, kui need indeksid on massiivi lõppu jõudnud. Näiteks, kui indekseid vähendatakse ja vasakpoolseim indeks on 0, siis selle uus väärtus on massiivi kõige suurem indeks. Lõpuks täida “display2” protsessi 15. lüliti olek. Seal peab igale numbrile kirjutama vastava “display_array”-s oleva väärtuse.

(5 punkti)

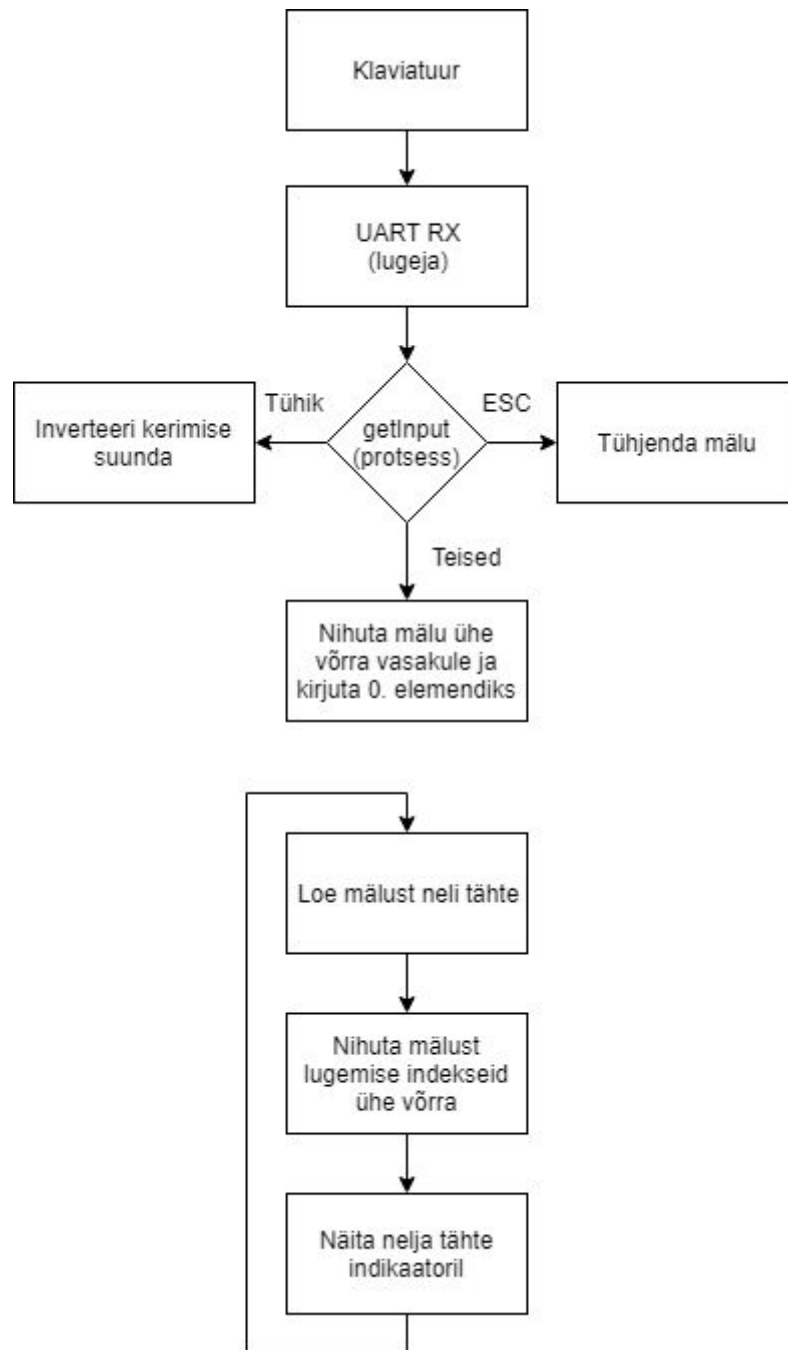
Ülesanne 3: Protsessis “getInput” loo funktsionaalsus kui lastakse lahti nupust “ESC”, siis tühjendatakse mälu, ja kui lastakse lahti tühikust, siis muudetakse kerimise suunda.

(1 punkti)

Kokku 15 punkti.



Joonis 1. Protsesside vahelised signaalid



Joonis 2. Loogikaskeem

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Robert Allik

1. Annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose **“Pmod lisamoodulite kasutamine “Digitaalse loogika” aine praktikumides”** mille juhendaja on Margus Rosin reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace'i kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Robert Allik

20.05.2019